

ربط الأكسيس مع الفيجوال بسيك من الصفر

إنشاء قاعدة بيانات أكسيس
مفهوم تحليل النظم والعلاقات بين الجداول
مفهوم ADO.Net
الفرق بين الوضع المنفصل والوضع المتصل
لغة الاستعلام عن البيانات SQL
إضافة، تعديل، حذف، بحث، عرض البيانات
مجموعة من الشفرات المفيدة

خالد السعداني

www.learn-barmaga.com



ربط الأكسيس
مع الفيجوال بسيك
من الصفر

من إعداد : خالد السعداني



بسم الله الرحمان الرحيم

بسم الله والصلاة والسلام على رسول الله، وعلى آله وصحبه ومن والاه وبعد،

فإن هذه الصفحات المعروضة بين يديك أخي الكريم / أختي الكريمة، تم إصدارها قبل سنتين ضمن سلسلة خطوة إلى الأمام التي كنا قد اعتزمنا على إيصالها إلى عشر خطوات، لكننا توقفنا عند الخطوة الثانية ولم نكمل بعدها لأننا انشغلنا بالكتابة في مسائل أخرى، فقدر الله بعد مدة أن أفتح هذا الملف لأعدله وأنقحه في نسخة جديدة ليلبي حاجة المتعلمين، لأن النسخة السابقة كانت تعرض صور البرامج المستعملة باللغة الفرنسية، والأصل أن هذه اللغة ليست متداولة في باقي الأقطار العربية باستثناء بلدان المغرب العربي لذلك كان من الجيد استبدال الصور الفرنسية بصور إنجليزية لأن ذلك أوضح وأبلغ.

بالإضافة إلى تنسيق الأكواد، فقد قمنا بإلغاء الخلفية التي كنا نضعها خلف الأكواد، وقمنا بتكبير حجم خط الكود ليكون واضحا وقابلا للنسخ بسهولة.

هذا وإن موضوع هذا الكتاب هو قواعد البيانات في الفيجوال بسيك دوت نيت، بحيث يعرض الفصل الأول طريقة إنشاء قواعد البيانات والجداول والعلاقات في ميكروسوفت أكسيس، موضحا الفرق بين أنواع العلاقات الممكن إنشاؤها بين الجداول.

بينما يعرض الفصل الثاني مختصرا للغة الاستعلام عن البيانات المشهورة SQL، ثم نبدأ رحلتنا العملية عبر عمليات الربط بين البرنامج وقاعدة البيانات، بالإضافة إلى أوامر الإضافة والتعديل والحذف والبحث والتنقل بين البيانات بطريقتين مختلفتين.



في حين يبقى الفصل الأخير مجرد ملحق، لمن يريد التعرف على كيفية إنجاز بعض المهام برمجيا، وقمنا بإدراج كود كل مشكلة مردفيه بشرح موجز.

وتجدر الإشارة إلى أنه يوجد كتاب آخر مماثل لهذا الكتاب الموجود بين يديك، وهو يعرض نفس الأمثلة المدروسة لكن مع قاعدة بيانات من نوع Microsoft SQL Server، وهو يحمل نفس اسم الكتاب ومتاح للتحميل على أكاديمية المبرمجين العرب.

هذا ونسأل الله عز وجل الإخلاص والصدق والتوفيق والسداد لنا ولسائر المسلمين، وأن يجعل هذا العمل المتواضع خالصا لوجهه الكريم، وأن لا يجعل للنفس فيه حظا.

خالد السعداني 27/10/2013

"يا أيها الذين آمنوا اتقوا الله و قولوا قولا سديدا. يصلح لكم أعمالكم و يغفر لكم ذنوبكم ومن يطع الله و رسوله فقد فاز فوزا عظيما"

الأحزاب : 70 و 71



للمراسلة من أجل طرح الاقتراحات والملاحظات والتساؤلات
نقدم لكم عنواننا الإلكتروني :

Khalid_ESSAADANI@Hotmail.fr

يسمح بطباعة كل كتب السلسلة إذا كان الغرض تعليميا وليس تجاريا

للتواصل المباشر أقدم لكم رقم هاتفي، فقط إذا كان الأمر مُلِحًّا جازاكم الله خيرا:

0673-07-51-05



بسم الله على بركة الله



الجزء الأول:

قاعدة البيانات

Database

Base de Données



تعتبر قواعد البيانات Database من أهم المصطلحات البرمجية وأكثرها ذيوعا، بحيث نجدها مرفقة مع معظم أنواع التطبيقات، ولعل ذلك راجع إلى أهميتها في حفظ البيانات وتنظيمها وتسهيل استعمالها.

وتنقسم قاعدة البيانات إلى نوعين:

- **محلية:** تستخدم من طرف مستخدم وحيد على جهاز واحد.
- **عن بعد:** وتكون مخزنة على أجهزة عن بعد، ويتم الولوج إليها عن طريق الشبكات Networks

وحتى يسهل التعامل مع البيانات يتم تخزينها على شكل جداول Tables، كل جدول يضم مجموعة من الحقول، والحقل هو عبارة عن حيز نقوم فيه بحفظ بعض البيانات، فمثلا لو افترضنا أننا بصدد إنشاء قاعدة بيانات للمدرسة، فسنحتاج إلى إنشاء مجموعة من الجداول مثل :

- **جدول التلاميذ:** وفيه نقوم بحفظ بيانات التلاميذ، الحقول الممكن إنشاؤها في هذا الجدول (الاسم، العنوان، السن، الجنس..).
- **جدول المواد التعليمية:** وفيه نقوم بحفظ بيانات المواد.
- **جدول النقط:** وفيه نقوم بحفظ النقط التي يقوم المعلمون بإدخالها.

طبعا هذا مجرد مثال تقريبي، فالجداول لا يمكننا إنشاؤها عشوائيا ولكن بعد دراسة وتحليل للمسألة، وتوجد مجموعة من لغات تحليل البيانات والتي تستعمل من أجل تفادي كل الأخطاء



المنطقية والمنهجية الممكن حدوثها على مستوى قاعدة البيانات، ومن بين تلك اللغات نذكر : UML,Merise...

إذن فقاعدة البيانات هي بمثابة مكتبة، والجداول هي بمثابة الرفوف، بحيث يتم حفظ البيانات في هذه المكتبة على رفوف خاصة بها.

للولوج إلى قاعدة البيانات والتحكم فيها يحتاج المستخدم إلى نظام إدارة قواعد البيانات، هنالك العديد من برامج إدارة قواعد البيانات، نسرد للمثال لا للحصر:

SQL Server, Oracle, Microsoft ACCESS, MySQL, Sybase...

هناك العديد من البرامج التي تخول لنا إنشاء قواعد البيانات والتعامل معها، ولعل أبرزها أوراكل Oracle، السيكويل سيرفر SQL Server ، ميكروسوفت أكسيس Microsoft Access ...

في هذا الدرس سوف ننشئ قاعدة البيانات بالاعتماد على ميكروسوفت أكسيس، ثم بعد ذلك نقوم بربط هذه القاعدة مع برنامج ننشئه بالفيجوال بزيك، وسوف نستعرض أغلب العمليات التي قد نحتاجها لإنشاء برنامج متكامل يعمل مع قواعد البيانات.

خلاصة:

- **قاعدة البيانات** هي مجموعة من المعلومات المخزنة بشكل منظم.
- **الجدول** عبارة عن مصفوفة ثنائية (أسطر وخانات)، يستعمل لحفظ البيانات.
- **نظام إدارة قواعد البيانات** هو برنامج يلعب دور الواجهة التي تربط المستخدم بقاعدة البيانات.



على بركة الله

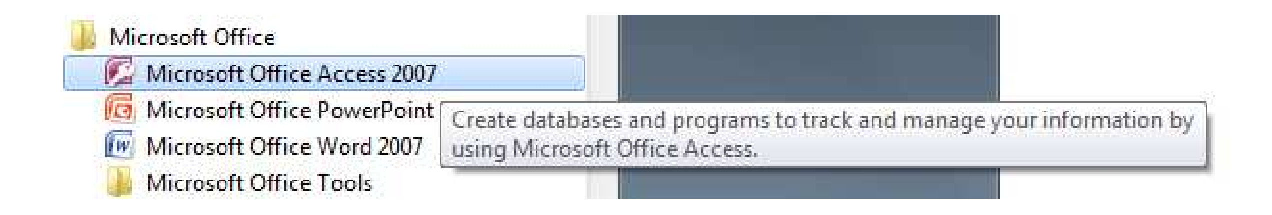
الآن إن شاء الله سوف ننتقل إلى برنامج Microsoft Access لنقوم بإنشاء قاعدة البيانات، ولكن قبل ذلك، دعني أوضح لك نقطة معينة:

ليس هنالك اختلاف على مستوى الشفرة البرمجية عند التعامل مع باقي أنظمة قواعد البيانات إلا في أشياء بسيطة، بحيث يبقى المبدأ هو هو، فالشفرة التي تقوم بربط البرنامج مع قاعدة البيانات من نوع أكسيس هي نفسها الشفرة التي تقوم بربط البرنامج مع SQL Server مع اختلاف بسيط تفرضه خصوصية كل نظام قواعد.

افتح برنامج ميكروسوفت أكسيس المنصب مع باقة ميكروسوفت أوفيس.

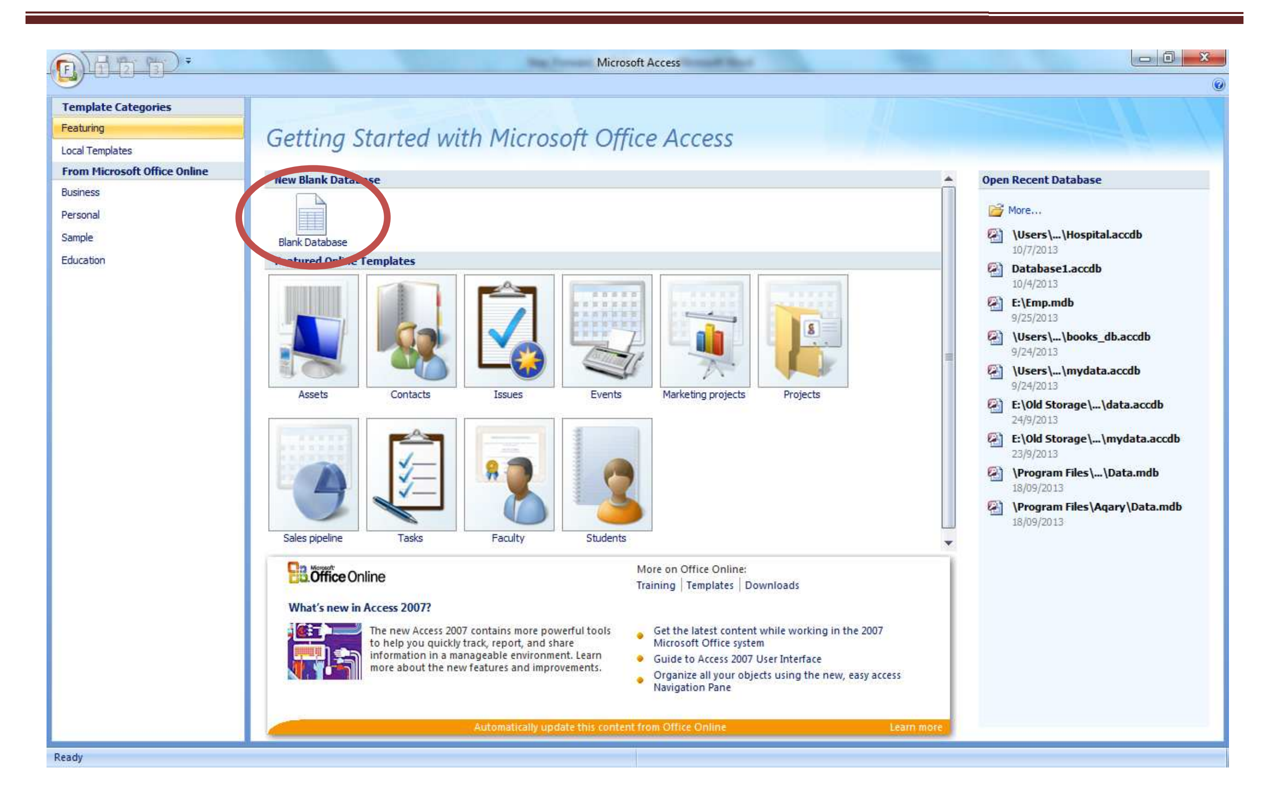
في مثالنا اليوم، سنشتغل على ميكروسوفت أكسيس 2007، إن كانت لديك نسخة أقدم فالأمر أسهل بكثير، وإن كانت لديك نسخة أحدث فالأمر سيان وليس هناك فرق كبير.

اذهب إلى القائمة Start، ثم All programs، ثم Microsoft Office، ثم Microsoft Office Access:



بعد فتح برنامج Access ستظهر لنا الشاشة التالية:

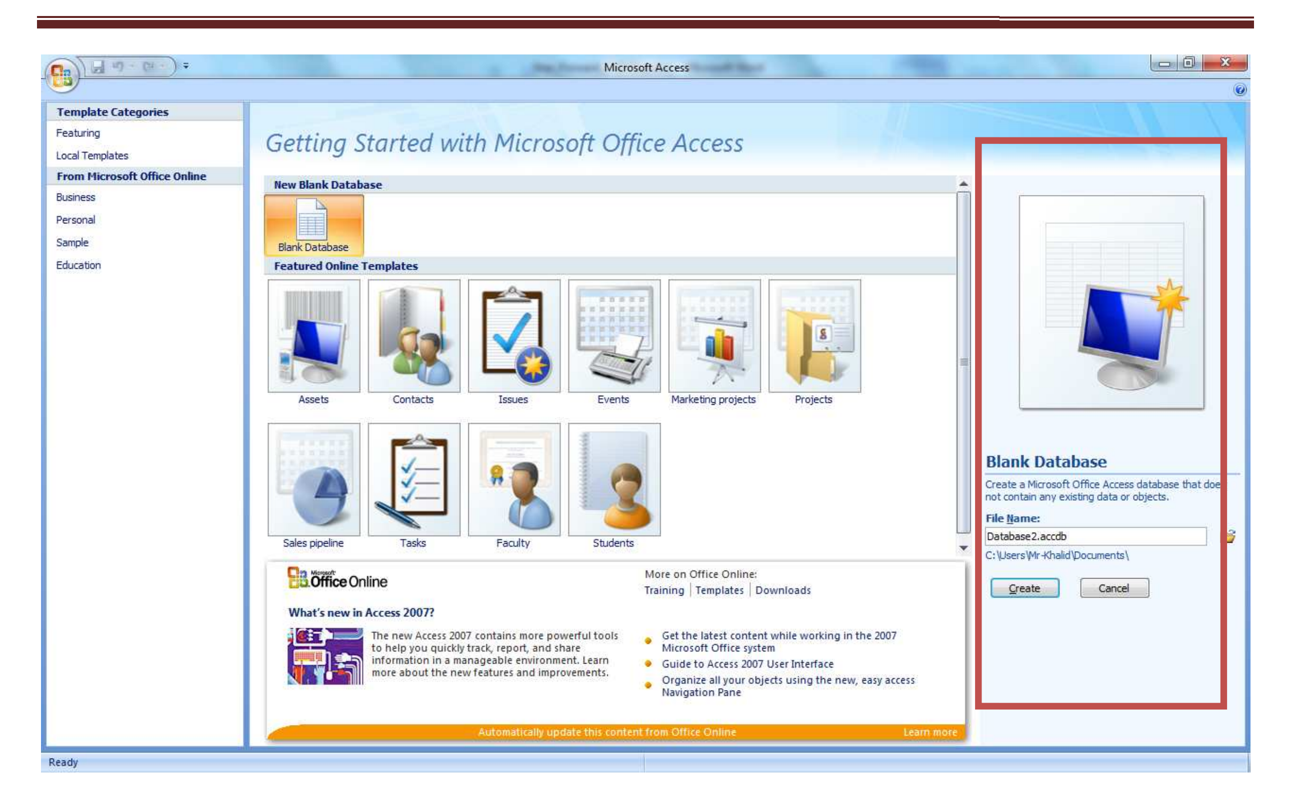




بعد ذلك اضغط على الأيقونة "قاعدة بيانات فارغة"، بالفرنسية "Base de données vide"، وبالإنجليزية "Blank Database".

ستظهر لك النافذة التالية فقم من خلالها بإعطاء اسم لقاعدة البيانات، واختيار مسار حفظها:

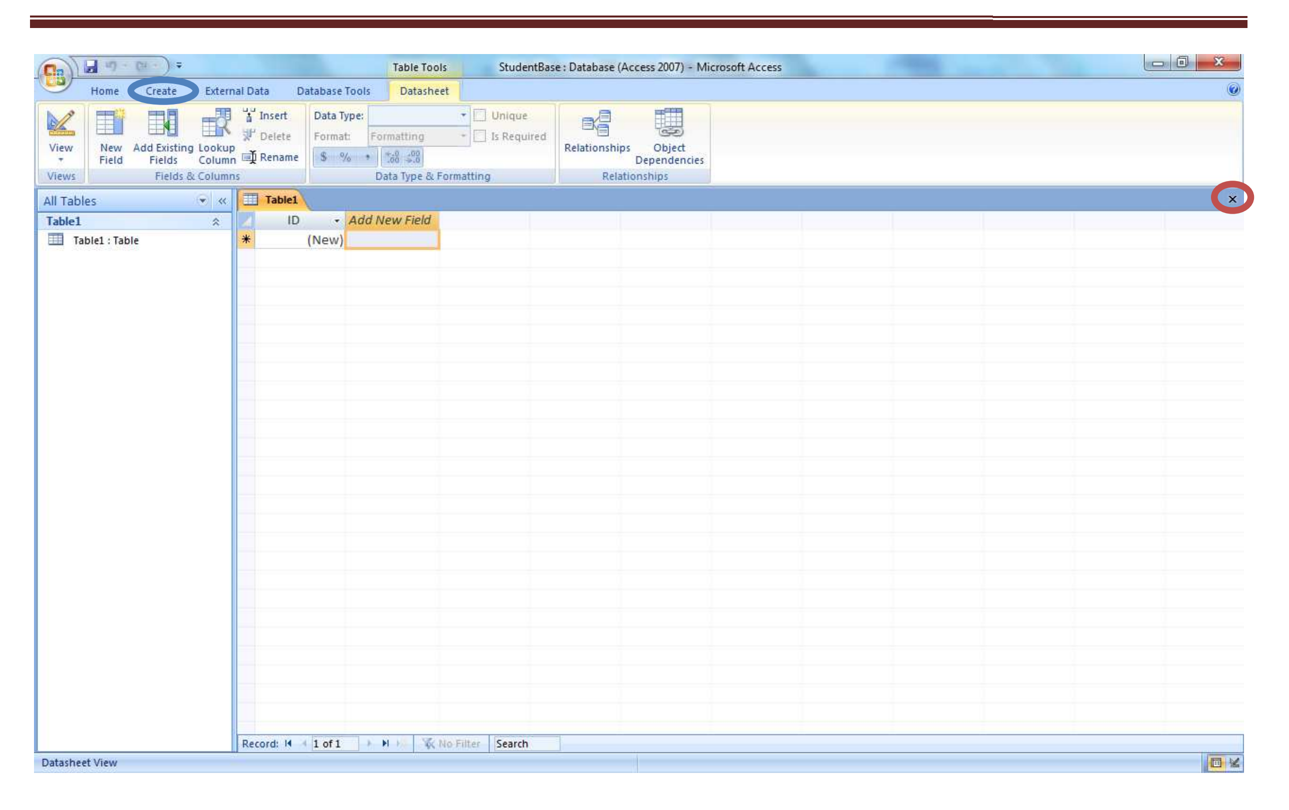




سننشئ قاعدة بيانات لإدارة الطلبة، فلنضع اسمها إذن StudentBase، أو ضع أي اسم يحلو لك فلا ديكتاتورية هنا ☺

ستلاحظ بعد الضغط على الزر إنشاء، بالفرنسية Créer، وبالإنجليزية Create، ظهور نافذة إنشاء الجداول التالية:





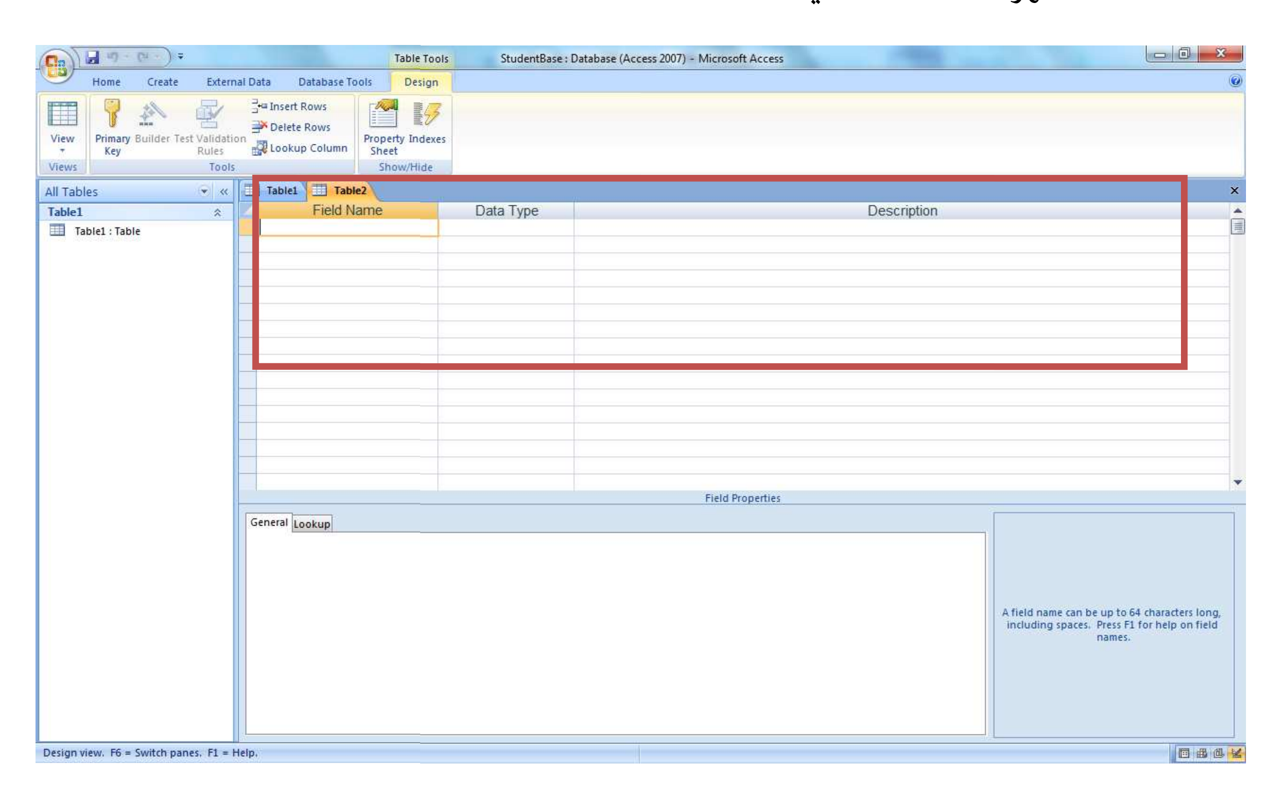
كما تلاحظ معي، هنالك جدول افتراضي يسمى Table1 نقوم بإغلاقه من زر الإغلاق المؤطر بالأحمر في المثال، ولنقم بعد ذلك بإنشاء جداولنا بأنفسنا.

من القائمة العليا اختر الأمر إنشاء (مؤطر بالأزرق في الصورة أعلاه)، ثم اختر الأمر "إنشاء جدول" كما في الصورة التالية:





عند الضغط ستظهر لك النافذة التالية:



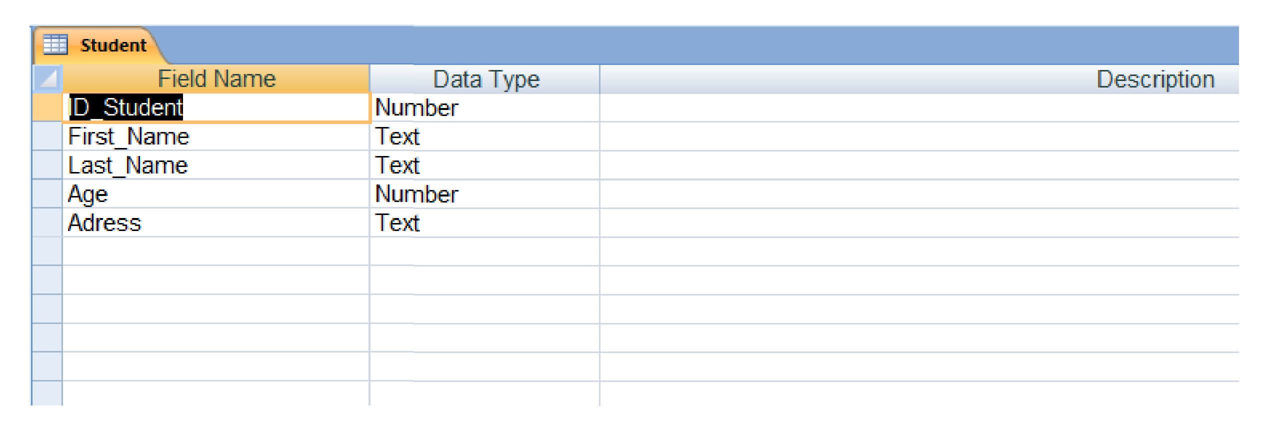
ما يهمنا هو المنطقة المحاطة باللون الأحمر، فكما ترى الجدول هو عبارة عن مجموعة من الحقول، لكل حقل نوع معين من البيانات ووصف خاص به، فمثلا لو أردنا إنشاء جدول الطلبة، فستكون الحقول كما يلي:

- رقم التسجيل: ويكون حقلا رقميا
- الإسم الشخصي: ويكون حقلا نصيا
- الإسم العائلي:ويكون أيضا حقلا نصيا
- بالإضافة إلى حقول أخرى كالسن والعنوان وغيرها.

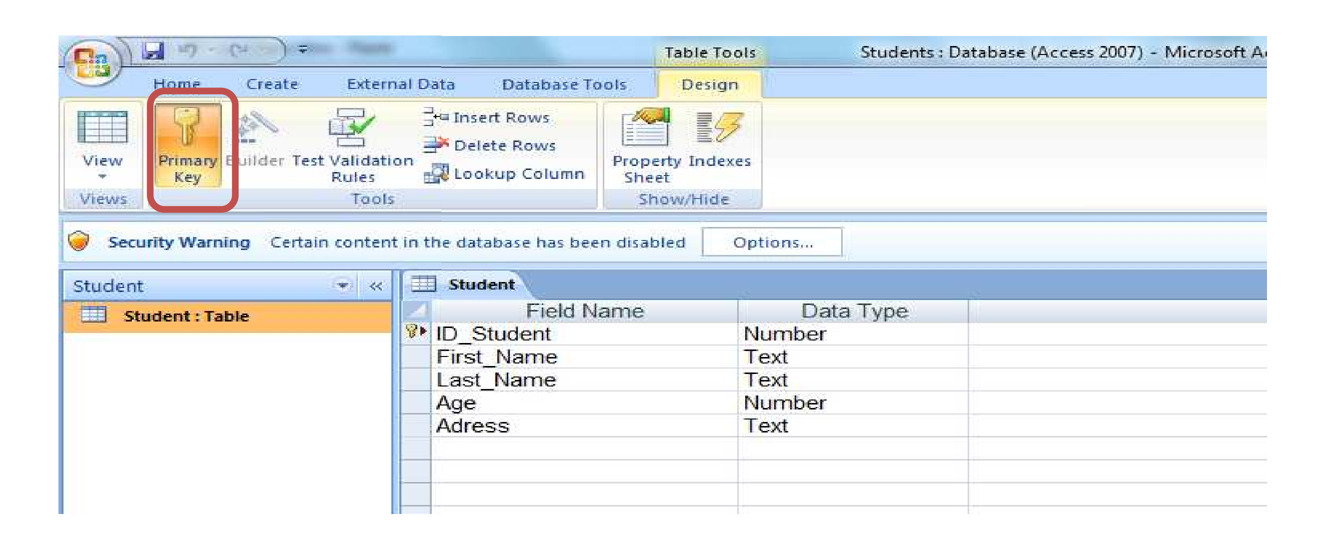


بالنسبة للوصف Description فلسنا مجبرين على كتابة أي شيء.

بعد هذه الإطلالة سوف نقوم إن شاء الله بإنشاء أول جدول لنا، وهو جدول الطلاب:



عند الانتهاء من تسجيل الحقول، سوف نذهب إلى الحقل الأول ID_Student ونضغط عليه بيمين الفأرة ونختار Primary key، بالفرنسية Clé Primaire، بالعربية أعتقد مفتاح أساسي، ويمكننا أيضا تحديده عن طريق القائمة التالية:





والغاية من جعل الحقل حقلا أساسيا، ، هو تفادي تكرار عنصر معين أكثر من مرة، فمثلا لا يجب أن نسجل طالبين بنفس الرقم وإلا حدث خلط على مستوى البيانات.

تعريف:

المفتاح الرئيسي دوره هو تحديد حقل معين يكون متفردا، لا يمكن تكرار قيمته، ولا يمكن أن تكون قيمته فارغة Null.

بعد ذلك نقوم بإغلاق الجدول، فتظهر نافذة تحذيرية تسألنا هل نريد أن نحفظ الجدول، نضغط على نعم، ثم نعطي للجدول اسما وليكن مثلا Student.

بنفس الطريقة السابقة ننشئ جدول المواد التدريسية Courses، وذلك بالذهاب إلى القائمة "إنشاء جدول" ثم نقوم بتسجيل الحقول التالية ونحدد رقم المادة على انه مفتاح أساسي Primary Key حتى لا يتم تكراره أكثر من مرة.

Student | Courses

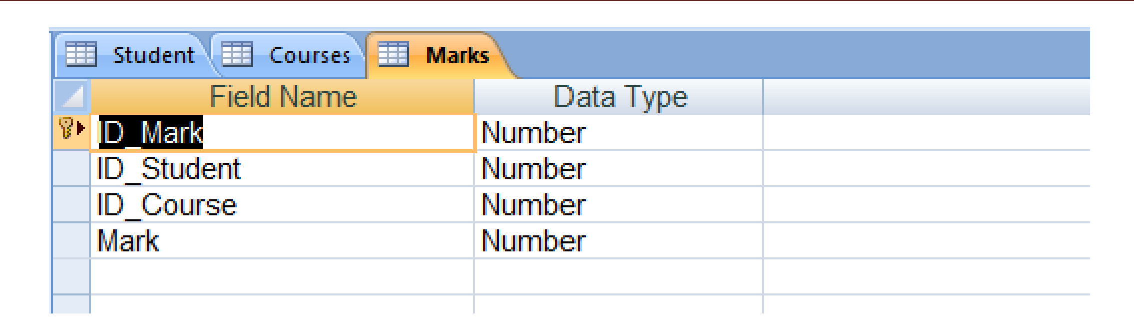
Field Name	Data Type
ID_Course	Number
Description	Text

كما تلاحظ فجدول المواد التدريسية يحتوي فقط على رقم المادة واسمها.

نقوم بحفظ الجدول باسم Courses.

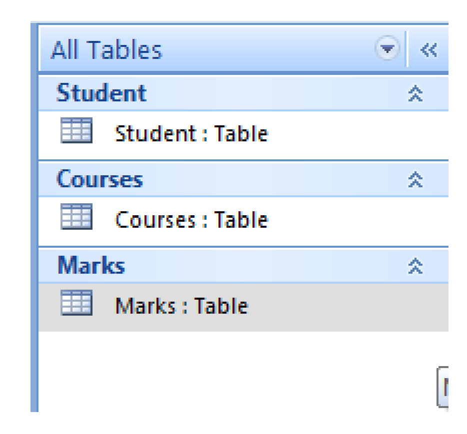
الآن سننشئ جدول النقط Marks وحقوله كما يلي:



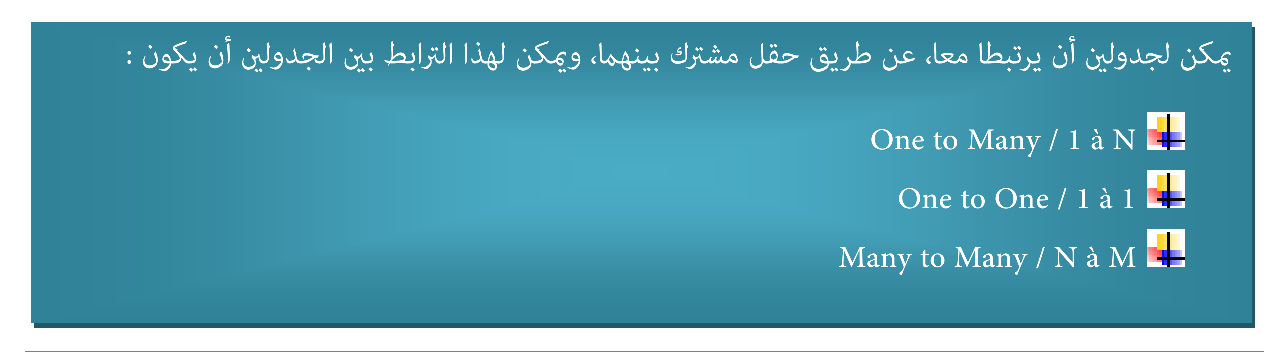


بالطبع لحفظ نقطة معينة، سنحتاج إلى اسم الطالب صاحب النقطة، وكذلك إلى المادة التي حصل فيها على هذه النقطة، قم بتحديد المفتاح الأساسي وأحفظ الجدول باسم Marks.

هذه هي الجداول التي قمنا بإنشائها:



العلاقات بين الجداول:



يمكن لجدولين أن يرتبطا معا، عن طريق حقل مشترك بينهما، ويمكن لهذا الترابط بين الجدولين أن يكون :

- One to Many / 1 à N
- One to One / 1 à 1
- Many to Many / N à M



علاقة واحد لأكثر (One to Many / 1 à N)

نستعمل هذه العلاقة عندما يكون عنصر **واحد متفرد** من جدول معين مرتبطا **بمجموعة** من العناصر في الجدول الآخر بواسطة حقل حاضر في الجدولين معا. ، وكمثال على ذلك نفترض أن لدينا جدول " السُّيَّاح" و جدول " الإقامة في فندق".

من المعلوم أن سائحا يمكن أن يحجز مجموعة من الإقامات، بحيث نجد أن كل إقامة حجزها هذا السائح تحتوي على الرقم الخاص به، بمعنى آخر لو أردنا الإطلاع على كل الإقامات التي قام بها السائح الفلاني، يكفي أن نبحث داخل جدول الإقامات عن المرات التي تكرر فيها رقمه.

تذكر جيدا:

> نستعمل هذه العلاقة عندما يكون عنصر **واحد متفرد** من جدول معين مرتبطا **بمجموعة** من العناصر في الجدول الآخر بواسطة حقل حاضر في الجدولين معا.

علاقة واحد لواحد (One to One / 1 à 1)

وهذه الحالة استثنائية، ونادرا ما نصادفها في تحليلنا لنظام ما، وتتمثل هذه العلاقة في كون عنصر متفرد واحد من أحد الجدولين مرتبط بعنصر متفرد واحد من الجدول الآخر بواسطة حقل يوجد في الجدولين معا.



وكمثال على ذلك نذكر جدولي "شراء سلعة" و "فاتورة الشراء"، بحيث نجد أن عملية شراء واحدة لها فاتورة واحدة خاصة بها، والعكس صحيح إذ لا يمكن لفاتورة واحدة لها رقمها الخاص أن تصحب أكثر من عملية شراء.

تذكر جيدا:

عنصر متفرد **واحد** من أحد الجدولين مرتبط بعنصر متفرد **واحد** من الجدول الآخر بواسطة حقل يوجد في الجدولين معا.

علاقة أكثر لأكثر (Many to Many / N à M)

وهذه العلاقة هي الأكثر ترددا وشيوعا، وكمثال لها الجداول التي تحدثنا عنها في أول الدرس، إذ أن العلاقة بين الطلاب والمواد التدريسية هي علاقة متعددة، بحيث يمكن للطالب **الواحد** أن يَدْرُسَ **مجموعة** من المواد، ويمكن للمادة **الواحدة** أن يَدْرُسَهَا **العديد** من الطلاب.

في هذه الحالة يجب أن يكون لجدول الطلاب حقل أساسي (Primary Key)، وكذلك جدول المواد، ثم نقوم بإنشاء جدول ثالث يضم الحقلين الرئيسيين معا، وفي حالتنا هذه هو جدول النقاط، إذ سنجده يضم رقم الطالب و رقم المادة وهما حقلان أساسيان في الجدولين المرتبطين.

تذكر جيدا:

العديد من عناصر الجدول الأول مرتبطة **بالعديد** من عناصر الجدول الثاني، لذا وجب إنشاء **جدول ثالث** يربط بينهما له حقلان هما المفتاحان الرئيسيان في الجدولين المرتبطين.

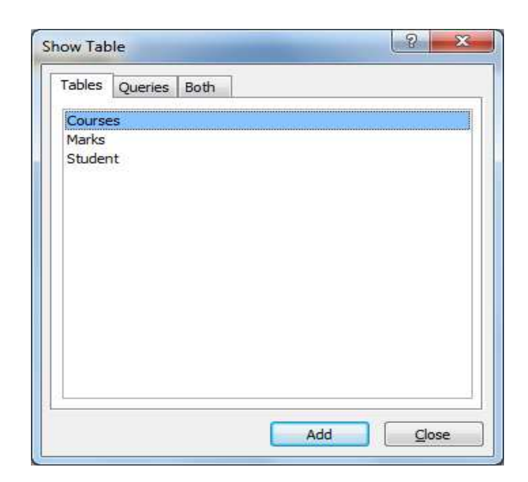


الآن انتهينا من إنشاء الجداول، بقي لنا فقط ربط هذه الجداول مع بعضها البعض، ولإنشاء العلاقات بين الجداول في ميكروسوفت أكسيس نذهب إلى القائمة العليا DataBase Tools، ثم نختار الأمر Relationships:



ستظهر لك النافذة التالي، وهي تضم كل جداول قاعدة البيانات التي تشتغل عليها حاليا، ومن خلال هذه النافذة يمكنك تحديد الجداول المراد إنشاء علاقات فيما بينها.

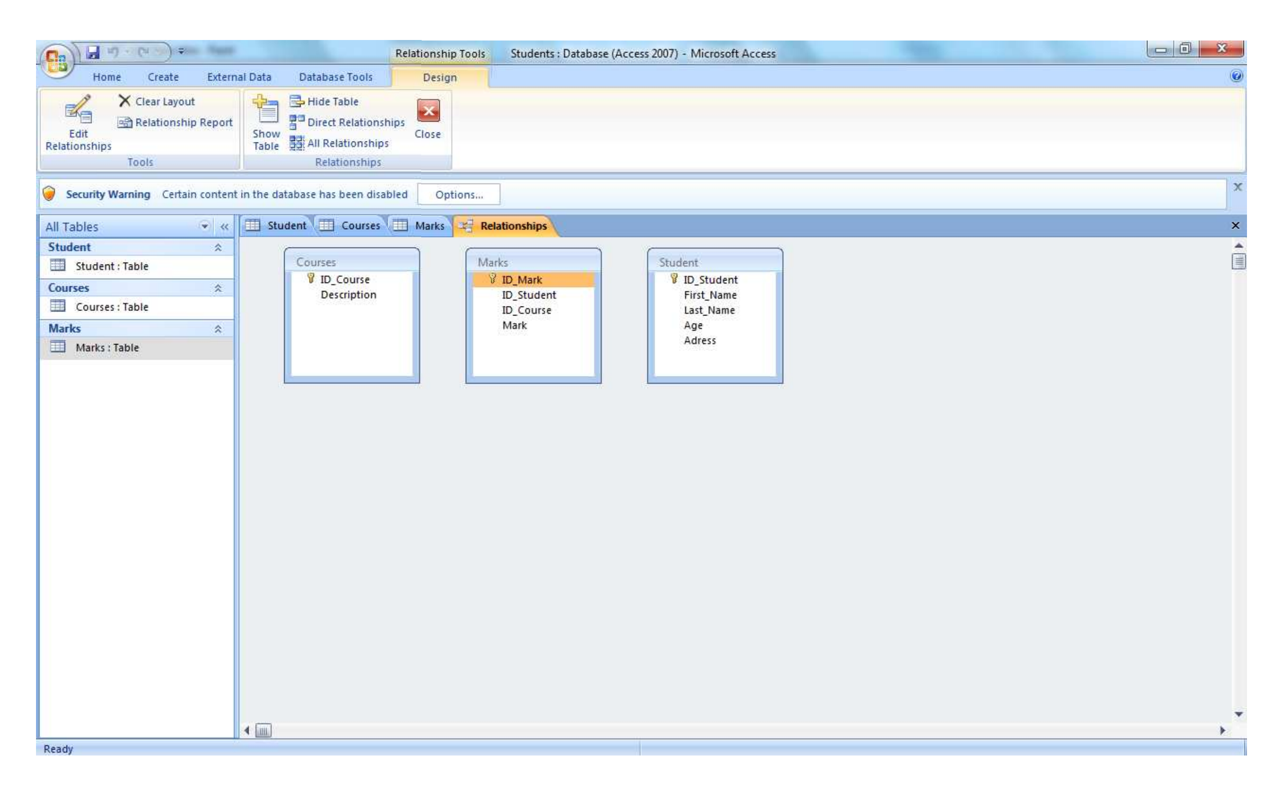
في مثالنا نحن سنختار الجداول الثلاثة للربط بينها:





ثم نضغط على الزر أضف، بالفرنسية Ajouter، وبالإنجليزية Add.

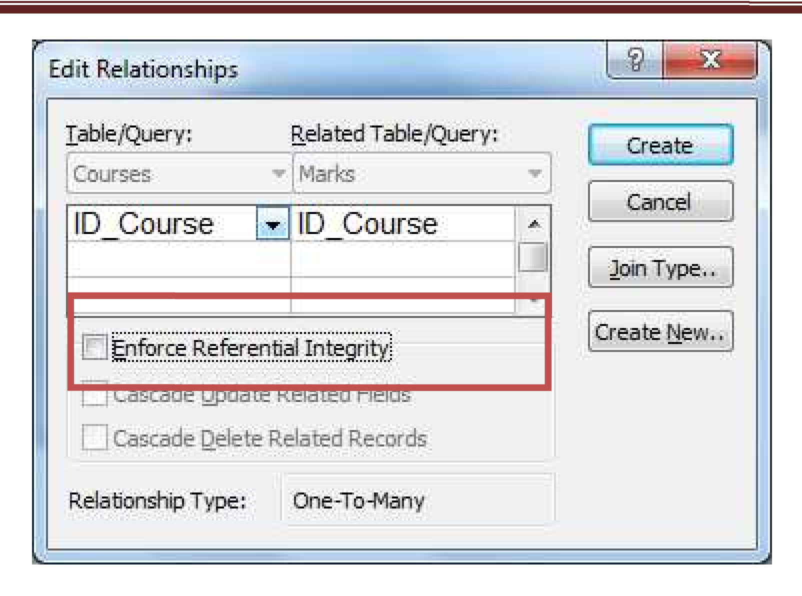
بعد ذلك ستظهر لك النافذة التالية:



لإنشاء علاقة بين جدولين، نضغط على المفتاح الأساسي ونجذبه إلى مثيله في الجدول الآخر، في حالتنا هذه سوف نذهب إلى الجدول Courses ونضغط على الحقل ID_Course، ونجذبه إلى الحقل ID_Course الموجود في الجدول Marks.

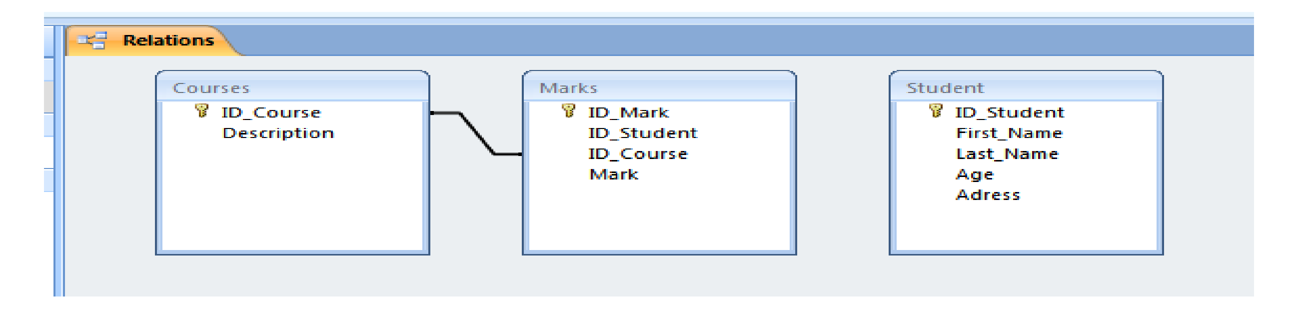
ستظهر لك هذه النافذة التالية:





بالنسبة للمنطقة المؤطرة بالأحمر، فتمكننا من تحديد ما إذا أردنا أن تحذف البيانات أو تُحدَّث في جدول عند حذفها أو تحديثها في الجدول الآخر المرتبط به.

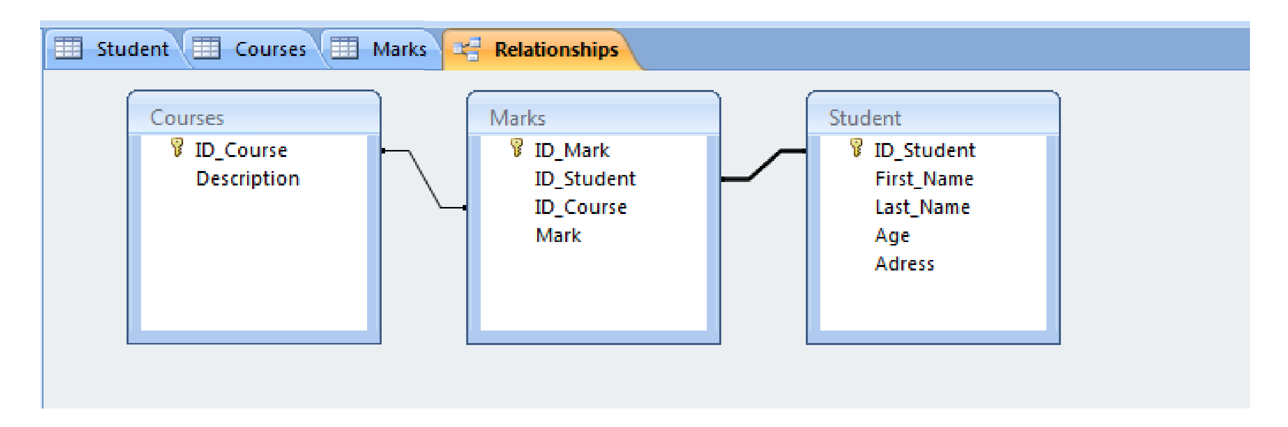
تأكد من الحقول المراد ربطها مع بعض، ثم اضغط على الزر إنشاء.



ستلاحظ أنه تم إنشاء خط وصل بين الجدولين في دلالة على أن العلاقة تمت بينهما، الآن نقوم بنفس الشيء مع الجدول Student ، نضغط على المفتاح الأساسي ID_Student ونقوم بجره إلى مثيله في الجدول Marks.

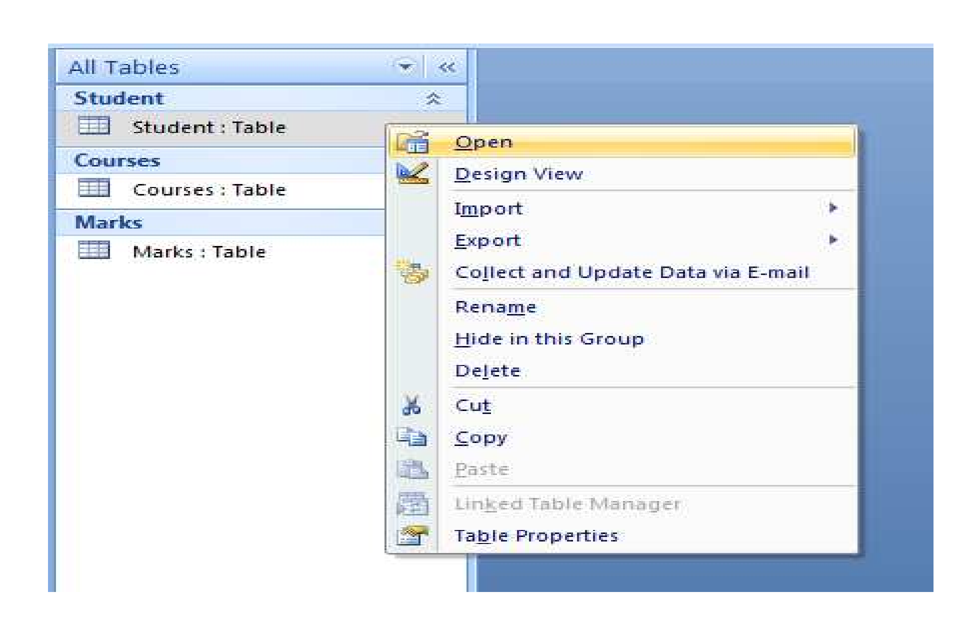


ليصبح الشكل النهائي للعلاقات بين الجداول كما يلي:



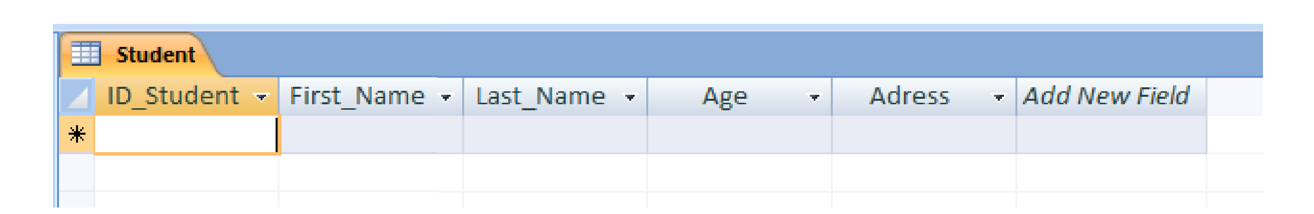
نقوم بحفظ العلاقات بالضغط على أيقونة الحفظ، أو بالضغط على Ctrl+S.

الآن لم يتبق لنا سوى ملء الجداول بقيم أولية، ولفعل ذلك نضغط على جدول الطلبة بيمين الفأرة، ونختار الأمر "فتح، بالفرنسية Ouvrir، وبالإنجليزية Open"



لتظهر لنا بعد ذلك النافذة التالية:

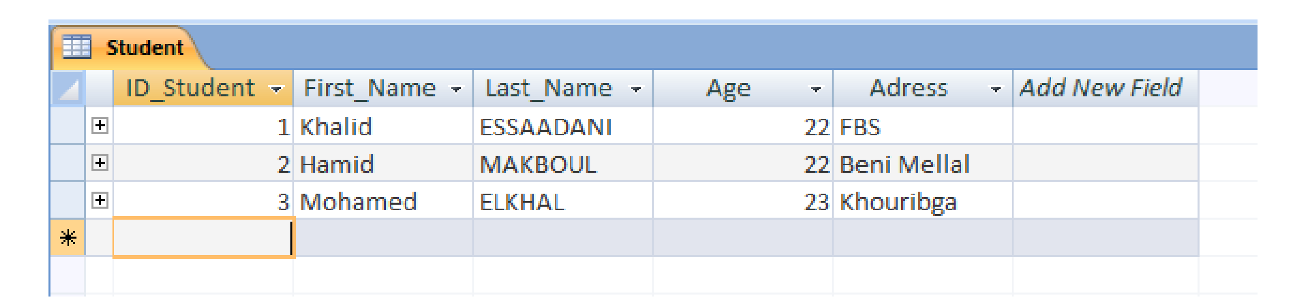




Student

ID_Student	First_Name	Last_Name	Age	Adress	Add New Field

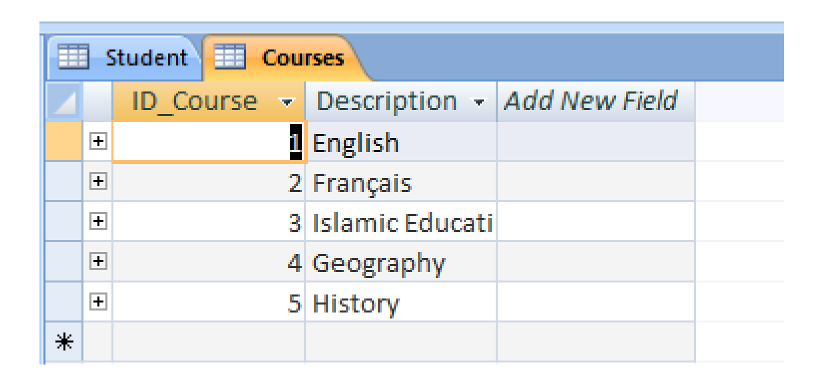
نقوم بتعبئة الجدول ببيانات اعتباطية، مثلا:



Student

ID_Student	First_Name	Last_Name	Age	Adress	Add New Field
1	Khalid	ESSAADANI	22	FBS	
2	Hamid	MAKBOUL	22	Beni Mellal	
3	Mohamed	ELKHAL	23	Khouribga	

نقوم بنفس الشيء بالنسبة لجدول المواد Courses:



Student | Courses

ID_Course	Description	Add New Field
1	English	
2	Français	
3	Islamic Educati	
4	Geography	
5	History	



الجزء الثاني:

ربط قاعدة البيانات مع الفيجوال استوديو



عموميات حول ADO.Net

كما عودناكم فلن نركز كثيرا على ماهو نظري، وإنما سنكتفي فقط بما تستلزمه الضرورة، لهذا سنعرض هنا بعض التعاريف والمعلومات النظرية المختصرة عن ADO.Net

في الفيجوال بسيك 6، حتى يتسنى التعامل مع قواعد البيانات نستعمل ADO و DAO، ولكن في الفيجوال بزيك.نيت، نستعمل ADO.Net وهي اختصار ل Activex Data Object .Net

وهي من بين التقنيات الجديدة والفعالة للتواصل مع قواعد البيانات، لأنها تتيح لنا الاشتغال مع مختلف أنواع قواعد البيانات بطريقة متشابهة مع تغيير طفيف على مستوى الشفرة.

و للتعامل مع أي مصدر بيانات، يلزمنا إقحام مجال الأسماء الخاص به، وسنعرض في الجدول التالي بعض مصادر البيانات الرئيسية وأمامها مجال الأسماء الخاص بها:

مجال الأسماء	مصدر البيانات
System.Data.SqlClient	للتعامل مع قواعد البيانات الخاصة ب Microsoft SQL Server
System.Data.OleDb	للتعامل مع قواعد البيانات الخاصة ب Microsoft ACCESS
System.Data.OracleClient	للتعامل مع قواعد البيانات الخاصة ب ORACLE



كما تعلم ـ عزيزي القارئ ـ فكل مجال أسماء يضم فئات منضوية تحته، فكذلك بالنسبة لمجالات الأسماء الخاصة ب ADO.Net، فهي تضم كائنات لكل منها دوره الخاص، وهي موحدة بالنسبة لكل مصادر البيانات أي أن نفس الكائنات سنحتاجها مهما كان نوع قاعدة البيانات فقط مع تغيير بسيط في اسم الكائن.

الكائنات الأساسية الخاصة ب ADO.Net هي كما يلي :

- Connection: دور هذا الكائن هو الربط بين التطبيق وقاعدة البيانات، إذ أنه يقوم بفتح اتصال بينهما.
- Command: يقوم بتنفيذ أمر (إضافة، تعديل، حذف..) على قاعدة البيانات
- Reader: يقوم بالقراءة من مصدر البيانات فقط (ReadOnly) ويُستعمل هذا الكائن في الوضع المتصل (On-line Mode)
- DataSet: وهي بمثابة مجموعة من البيانات، ويمكن أن تضم جداول وعلاقات كما هو الحال مع قاعدة البيانات، لها دور أساسي في تكديس البيانات في الذاكرة. ويُستعمل هذا الكائن في الوضع المنفصل(Off-line Mode)
- DataAdapter: له دور أساسي إذ يقوم بملء الداتاسيت بالبيانات، وكذلك يقوم بعمليات الإضافة والتحديث والحذف في مصدر البيانات.



مررنا قبل قليل على مصطلحين جديدين وهما: الوضع المتصل والوضع المنفصل، وهما وضعان مختلفان في ADO.Net للتواصل مع قواعد البيانات، لكل واحد مميزاته وكائناته، ويشتركان أيضا في بعض الكائنات الأساسية.

1. **الوضع المتصل**: عندما نربط برنامجنا بقاعدة البيانات في هذا الوضع فالمراحل التي نمر منها كما يلي:
   - نقوم بفتح الاتصال Connection مع قاعدة البيانات
   - نجري عمليات القراءة والكتابة على قاعدة البيانات مباشرة
   - نغلق الاتصال Connection
2. **الوضع غير المتصل (المنفصل)**: وفي هذا الوضع تصبح المراحل كما يلي:
   - نقوم بفتح الاتصال Connection
   - نضع نسخة من قاعدة البيانات أو جزء منها في الذاكرة (هل تتذكر ما قلنا حول الداتاسيت؟)
   - نغلق الاتصال Connection
   - نجري عمليات القراءة والكتابة على النسخة المحفوظة على الذاكرة
   - عندما ننتهي، نفتح الاتصال ونرسل المعطيات الجديدة إلى قاعدة البيانات الأصلية لتحديثها، ثم نغلق الاتصال.



لغة SQL:

لكي يتم التواصل مع قواعد البيانات، يستخدم ADO.Net لغة السيكويل SQL ، بالنسبة للإخوة الذين ليست لهم دراية بهذه اللغة، فهي لغة متخصصة في قواعد البيانات حيث تتيح لنا القيام بكل أنواع العمليات من إضافة وتعديل وحذف وغيرها على قواعد البيانات.

من أراد التعمق في لغة SQL، فقد أفردنا لها كتابا كاملا مكونا من زهاء 130 صفحة تحت مسمى "الشرح الوافي لتعلم لغة SQL من نبعها الصافي" يمكنك تحميله من أكاديمية المبرمجين العرب أو البحث عنه في محركات البحث لأنه منتشر بكثرة ولله الحمد.

لإضافة عنصر جديد إلى جدول، فصيغة الإضافة تكون كما يلي:

```
Insert into Table1 (Field1,Field2) values (¨Value1¨,¨Value2¨)
```

مثال :

إذا أردنا أن نضيف معلومات طالب جديد فسيكون الاستعلام كما يلي:

```
Insert into Student (ID_Student,First_Name,Last_Name,Age,Adress) values (¨1¨,¨Khalid¨,¨ESSAADANI¨,22,
¨Fkih Ben Salah¨)
```

لحذف عنصر من جدول، فالصيغة كما يلي:

```
Delete * From Table1 Where Field1="Value1"
```

ويعني هذا الاستعلام: قم بحذف كل الأسطر التي تحتوي على الحقل Field1 ذو القيمة Value1 ، النجمة * تعني "جميع".



مثال:

سنقوم بحذف كل الطلبة الذين يفوق عمرهم 20 سنة:

```
Delete * From Student Where Age > 20
```

لتعديل عنصر في جدول معين، فالصيغة هكذا:

```
Update Table1 Set Field1="Value1", Field2= "Value2 ",...
```

مثال:

سنقوم بتغيير الاسم الشخصي للطالب صاحب الرقم 2:

```
Update Student Set First_Name="Habib" Where ID_Student=2
```

لجلب البيانات من جدول أو مجموعة من الجداول، فالصيغة كما يلي:

```
Select Field1, Field2 From Table1
```

مثال:

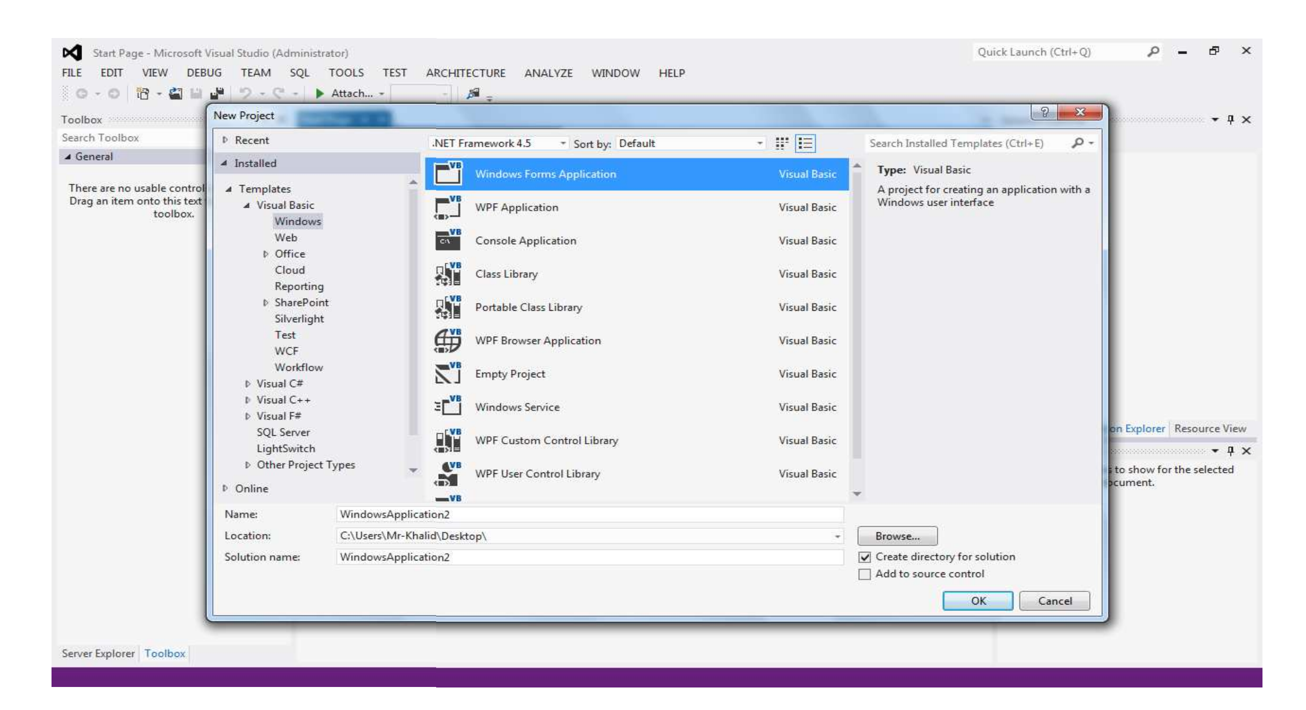
سنقوم بجلب بيانات كل التلاميذ الذين يسكنون في مدينة الرياض:

```
Select * From Student Where Adress= "Riad"
```



التطبيق

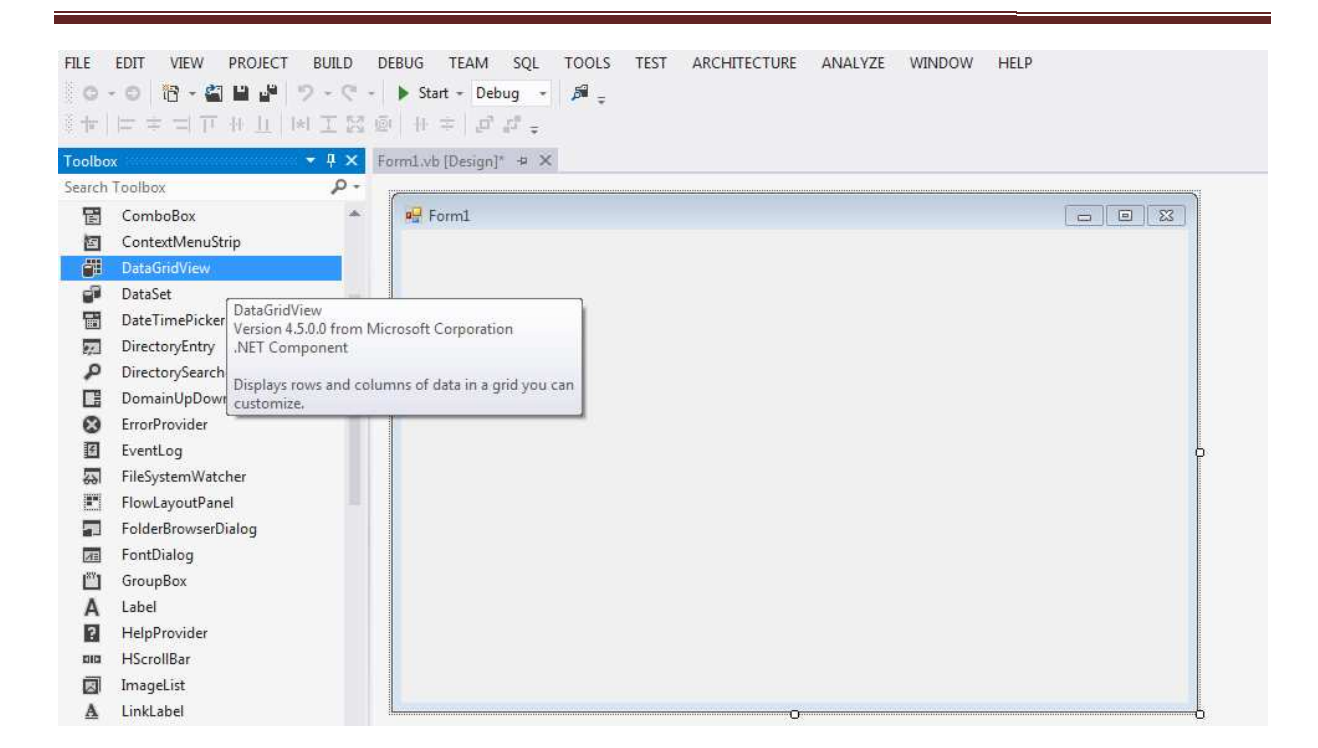
افتح مشروعا جديدا في الفيجوال استوديو من نوع Windows Forms Application، سمه كما تريد:



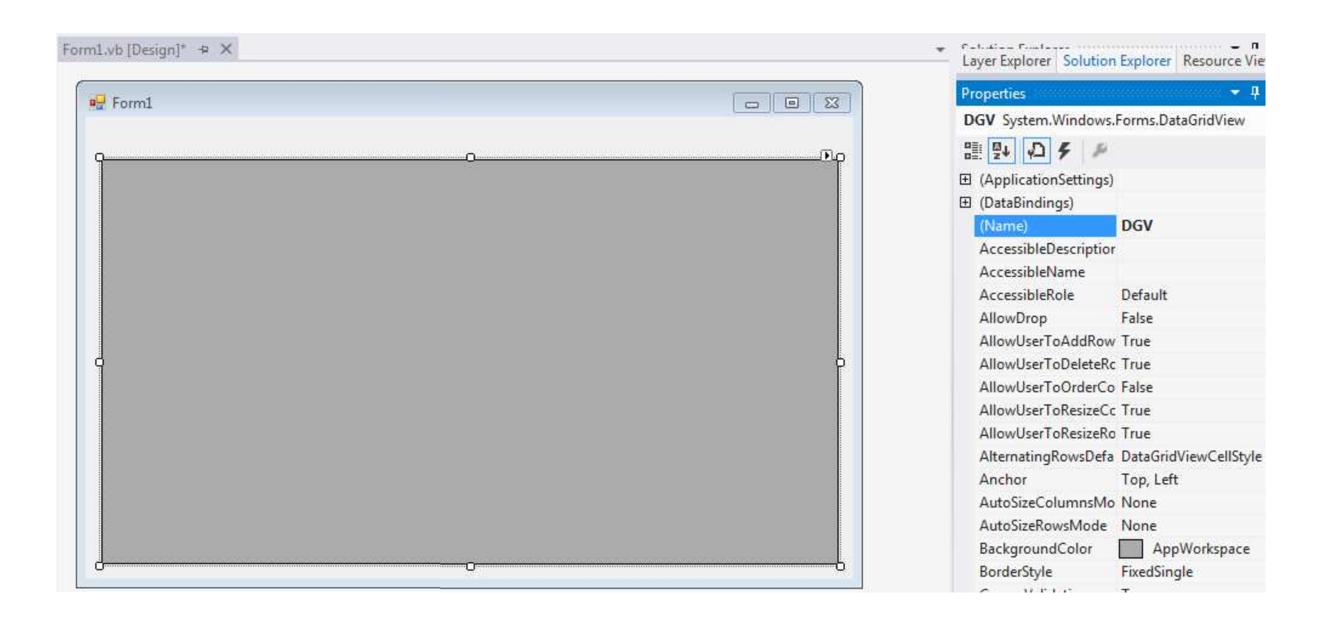
سوف نقوم بعرض لائحة تضم كل الطلبة المسجلين عندنا في قاعدة البيانات، وللقيام بذلك إذهب إلى علبة الأدوات وقم بالبحث عن DataGridView ثم قم بجذبها إلى الفورم.

DataGridView:

هي عبارة عن أداة تقوم بعرض البيانات على شكل جدول (أسطر و أعمدة)، كما تمكننا أيضا من إجراء عمليات الإضافة والحذف والتعديل عليها دون الحاجة إلى استعمال أدوات أخرى



قم بتغيير اسم DataGridView إلى DGV:





الآن قم بالضغط مرتين على الفورم لننتقل إلى نافذة الكود:

```
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub
End Class
```

أول شيء سنقوم به، هو جلب مجال الأسماء Namespace الذي سيتيح لنا التعامل مع قاعدة البيانات من نوع Access، وهو System.Data.OleDb ، ويختلف هذا المجال حسب نوع برنامج قواعد البيانات، فمثلا بالنسبة لل SQL Server فمجال الأسماء هو:System.Data.SqlClient

إذن سوف نقوم بالذهاب إلى أول سطر في نافذة الكود، ونجلب مجال الأسماء الخاص بقواعد البيانات من نوع Access.

```
Imports System.Data.OleDb
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal
    e As System.EventArgs) Handles MyBase.Load

      End Sub
End Class
```

بعد ذلك نقوم بإنشاء الاتصال مع قاعدة البيانات، ولفعل ذلك نعلن على متغير من نوع OleDbConnection.




```
Imports System.Data.OleDb
Public Class Form1

    Dim Con As New
    OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
    Source= StudentBase.accdb")

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal
    e As System.EventArgs) Handles MyBase.Load

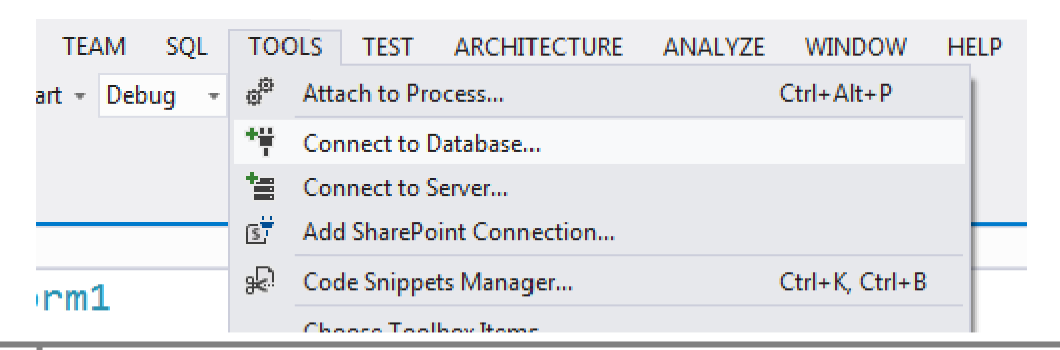
     End Sub
End Class
```

كما ترى فنص الاتصال ConnectionString له مجموعة من البارامترات وهي كما يلي:

Provider: وهو مزود البيانات الذي نشتغل عليه، وفي حالتنا هذه هو ميكروسوفت أكسيس ذو النسخة 12.0

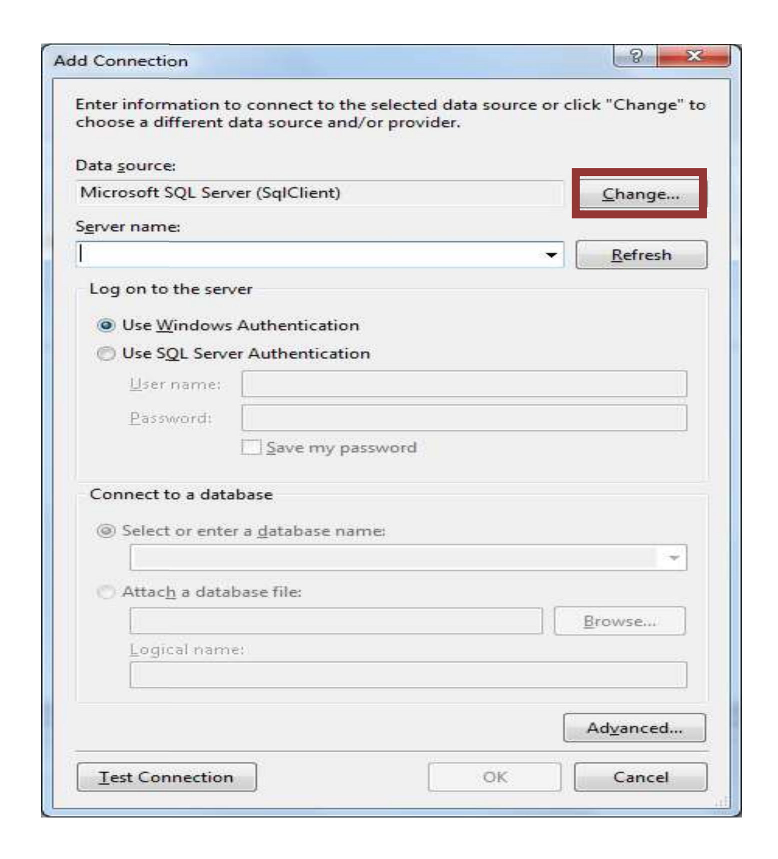

Data Source: وهو المسار الذي توجد عليه قاعدة البيانات.

ويمكنك الحصول على نص الاتصال ConnectionString عن طريق الذهاب إلى القائمة Tools ثم اختيار القائمة الفرعية Connect to database





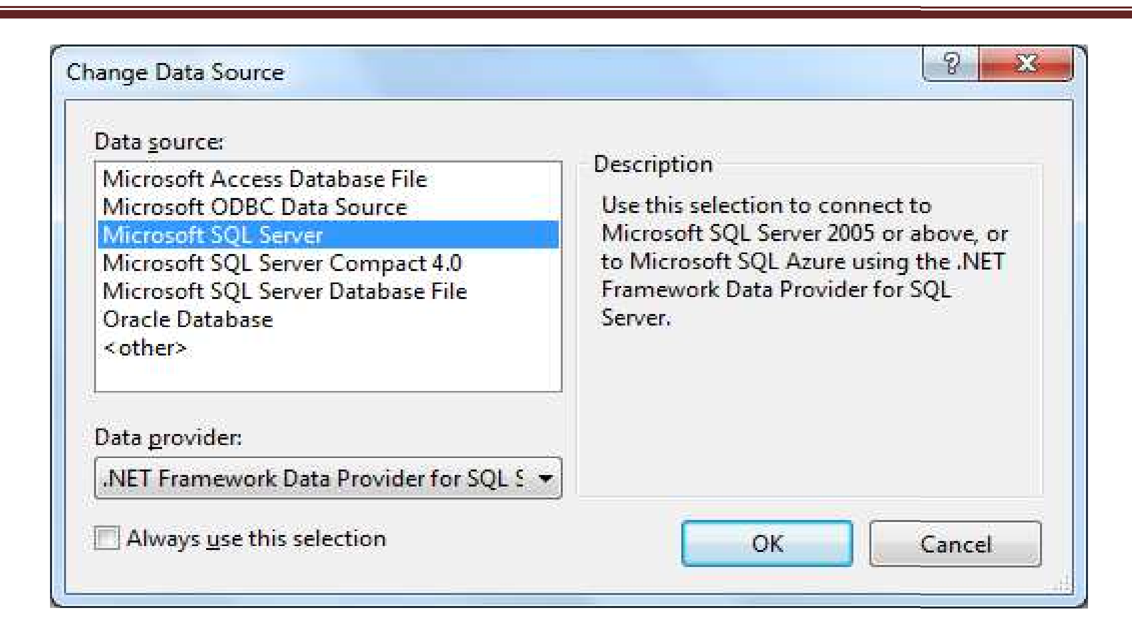
سوف تظهر لك النافذة التالية:



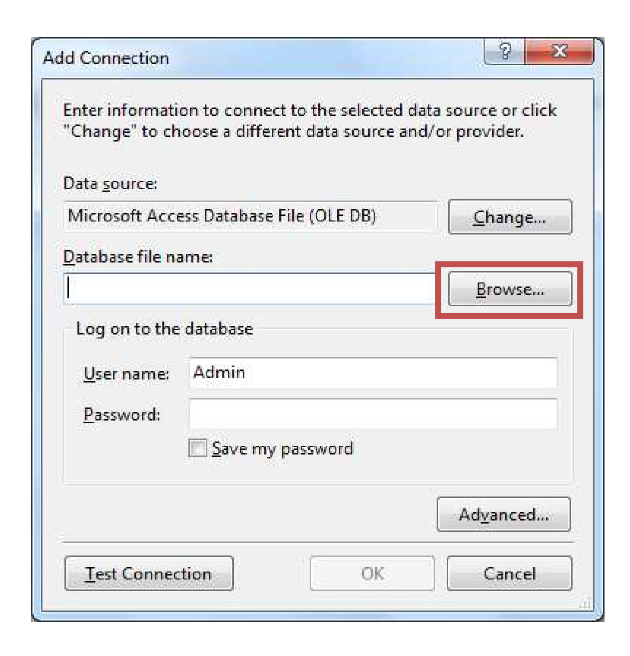
مصدر البيانات الافتراضي هو Microsoft SQL Server، لذلك علينا استبداله بميكروسوفت أكسيس عن طريق الضغط على الزر Change المؤطر بالأحمر في الصورة أعلاه.

بعد الضغط على الزر Change ستطالعك النافذة التالية:





قم بتحديد الخيار الأول Microsoft Access Database File واضغط على OK، لتظهر لك النافذة التالية:

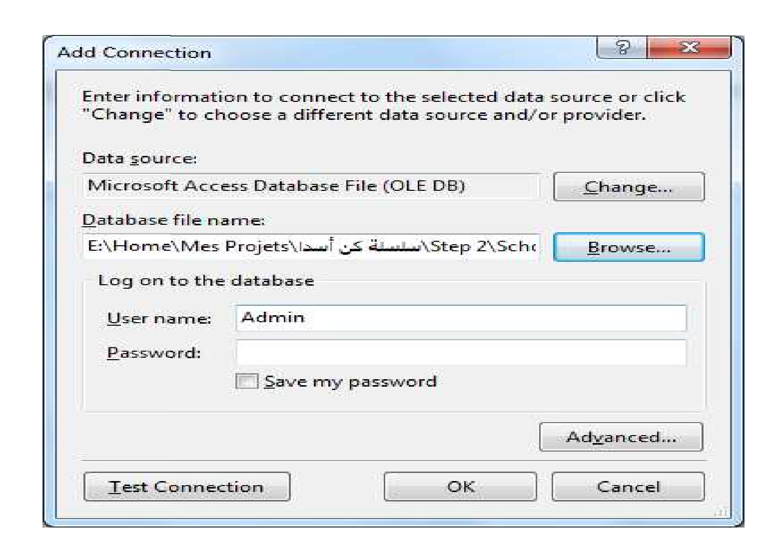


قم بالضغط على الزر Browse وحدد مكان حفظ ملف قاعدة البيانات:



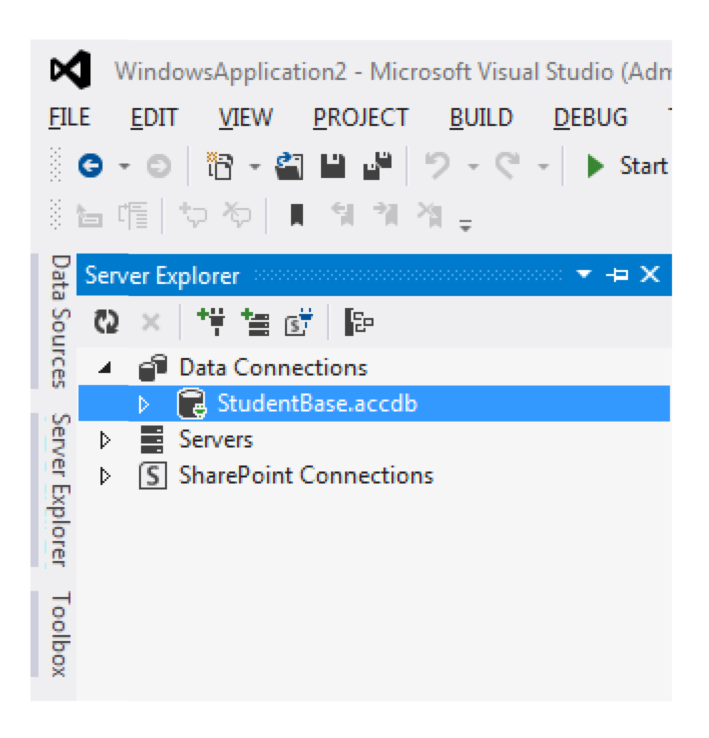


حدد ملف قاعدة البيانات واضغط على Open، ستظهر لك نفس النافذة السابقة، لكن هذه المرة تطلب منك كتابة اسم المستخدم وكلمة المرور إذا كانت قاعدة البيانات محمية ببيانات الدخول:

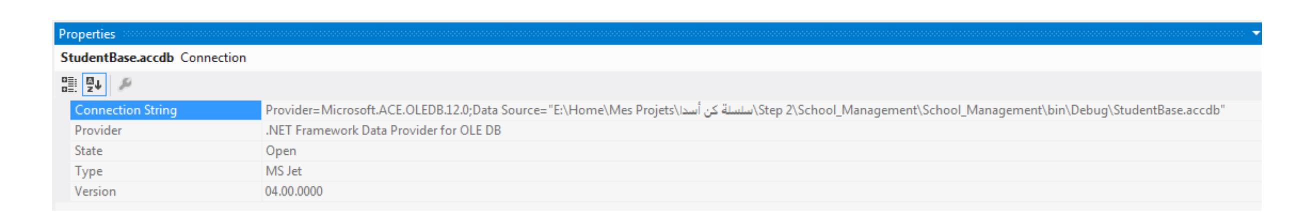




إذا لم تكن قاعدة البيانات تتوفر على كلمة مرور قم بالضغط على OK مباشرة، لتتم إضافة قاعدة البيانات إلى متصفح السيرفرات Server Explorer، الذي يظهر في الصورة التالية:

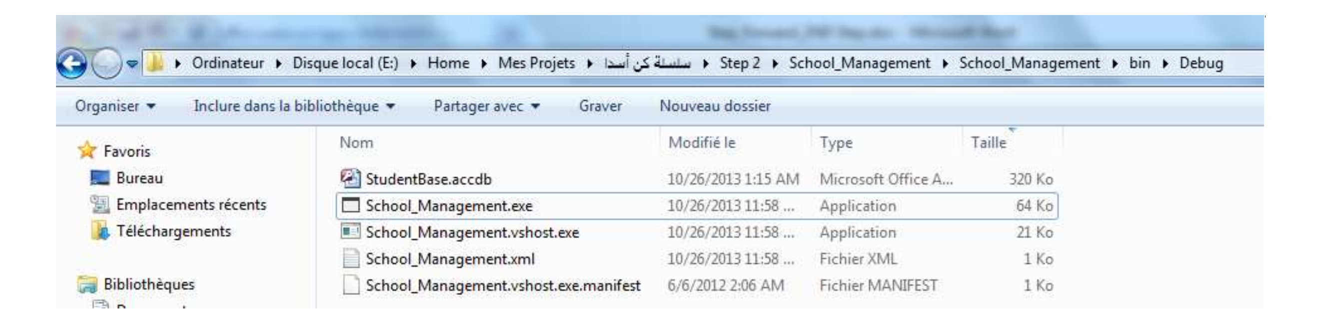


الآن تستطيع مشاهدة نص الاتصال، بعد أن تحدد ملف قاعدة البيانات في Server Explorer، واضغط على F4 من لوحة المفاتيح لتظهر لك نافذة الخصائص، أو اضغط بيمين الماوس على قاعدة البيانات وقم باختيار Properties.





قيمة نص الاتصال هي ذلك النص المكتوب أمام الخاصية Connection String ، لاحظ أن مسار ملف قاعدة البيانات طويل جدا، لذلك يمكنك تقصيره عبر وضع ملف قاعدة البيانات داخل مجلد الملف التنفيذي للمشروع Bin\Debug، أي في المسار التالي:



الآن سيصير بإمكاننا كتابة نص الاتصال كما يلي:

```
Dim Con As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=StudentBase.accdb")
```

بدل كتابتها بالشكل التالي:

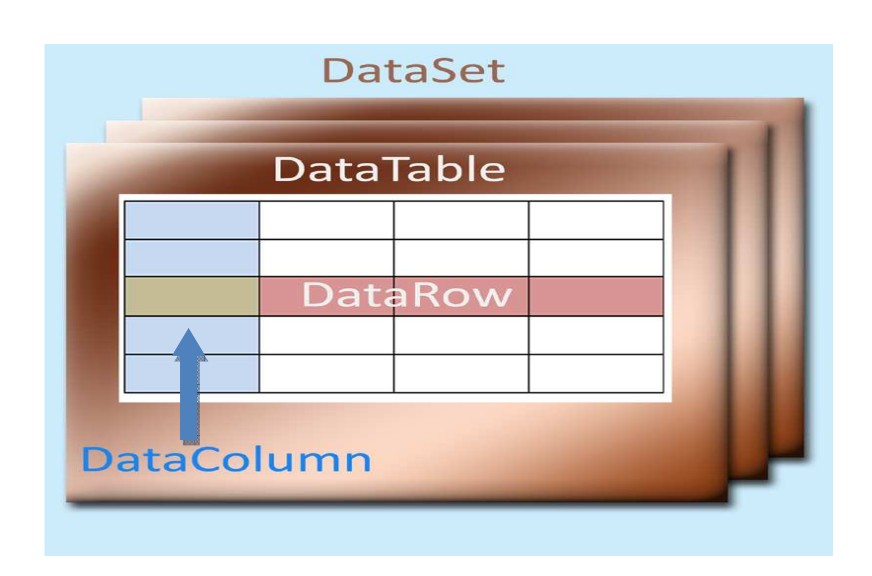
```
Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=E:\Home\Mes Projets\سلسلة كن أسدا\Step
2\School_Management\School_Management\bin\Debug\StudentBase.accdb")
```

الآن تعرفنا على طريقة الحصول على سلسلة الاتصال، فيما بعد ستكتبها بالاعتماد على نفسك لأنها سهلة الحفظ !



قمنا بإنشاء سلسلة الربط بين البرنامج وقاعدة البيانات، سنجلب الآن بيانات جدول الطلاب ونظهرها في الداتاغريدفيو، وللقيام بذلك سنحتاج إلى كائنين اثنين:

- OleDbDataAdapter: ودوره هو تنفيذ استعلامات SQL التي رأيناها سابقا.
- DataTable: وهي بمثابة جدول لتخزين ما سيقوم OleDbDataAdapter بجلبه، رأينا فيما سبق أن كائن الداتاسيت DataSet هو بمثابة مجموعة من الجداول، أما DataTable فهو جدول واحد، بمعنى آخر يمكننا القول أن DataSet هي مجموعة من DataTable، و هذه الأخيرة عبارة عن جدول فيه أسطر DataRow و خانات DataColumn ويمكننا استعمال الكائن الذي نريد:



في مثالنا هذا سنشتغل على DataTable.



أي أن شفرة الإعلان ستكون كما يلي:

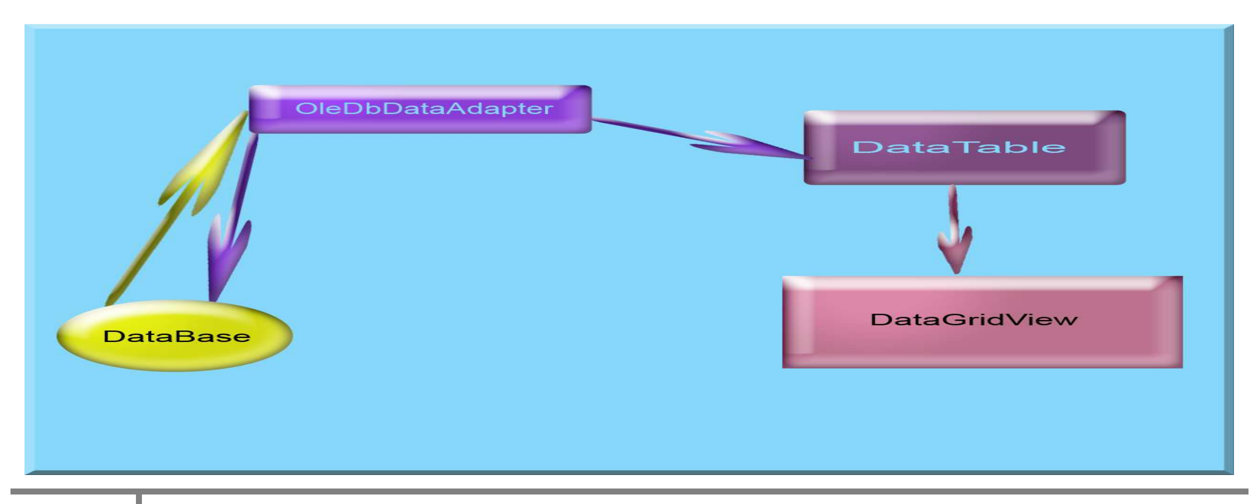
```
Imports System.Data.OleDb
Public Class Form1
    Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source= StudentBase.accdb")

    Dim Da As OleDbDataAdapter
    Dim Dt As New DataTable

    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub
End Class
```

الآن سنذهب إلى الحدث Form1_Load ونجلب بيانات جدول الطلاب باستعمال OledbDataAdapter ، ثم نخزن ذلك في DataTable، وفي الأخير نقوم بعرض البيانات في الداتاغريدفيو، كما توضح الصورة التالية:





نقوم بكتابة الشفرة التالية:

```
Imports System.Data.OleDb
Public Class Form1
    Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source= StudentBase.accdb")

    Dim Da As OleDbDataAdapter
    Dim Dt As New DataTable

    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

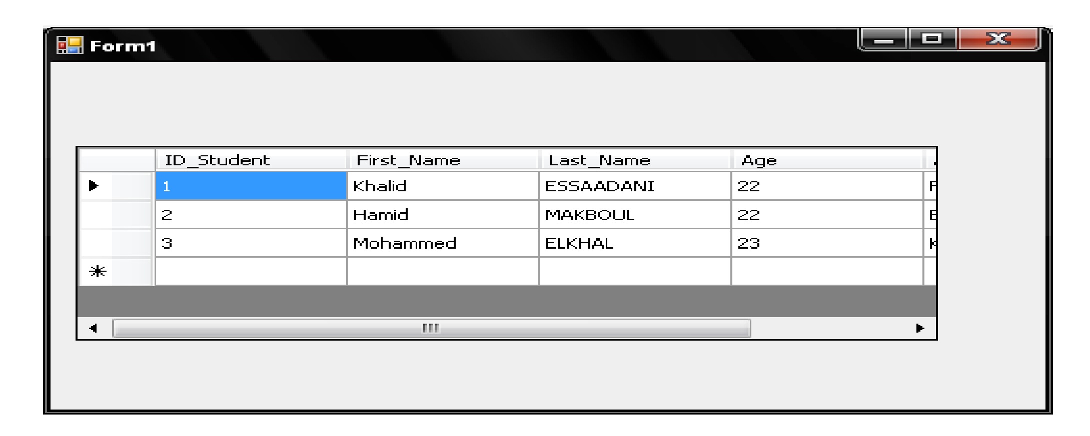
        Da = New OleDbDataAdapter("Select * From Student",
Con)
        Da.Fill(Dt)
        DGV.DataSource = Dt

    End Sub
End Class
```

في السطر الأول قمنا بإرسال استعلام جلب البيانات من القاعدة بالاعتماد على OleDbDataAdapter وحددنا له سلسة الاتصال Con التي ينبغي أن يمر عبرها، ثم بعد ذلك أفرغنا ما قام بجلبه في DataTable عن طريق الدالة Fill، وفي الأخير قمنا بملء DataGridView من هذه DataTable.

إذا قمنا بتنفيذ البرنامج فسوف نحصل على النتيجة التالية:





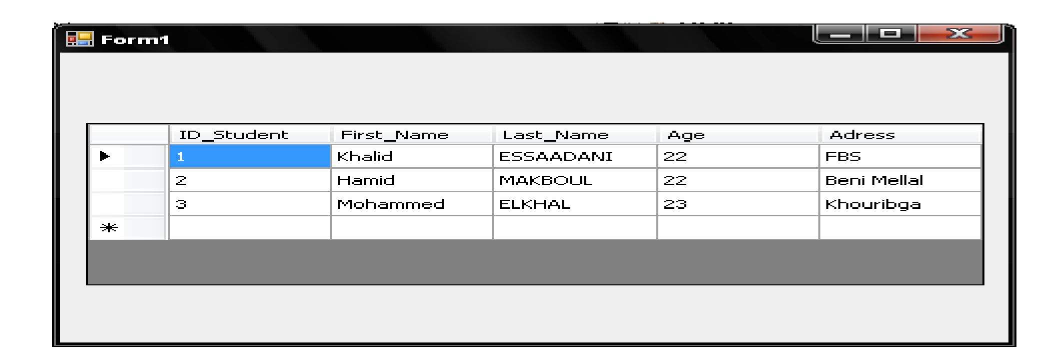
	ID_Student	First_Name	Last_Name	Age
▶	1	Khalid	ESSAADANI	22
	2	Hamid	MAKBOUL	22
	3	Mohammed	ELKHAL	23
*				

كما ترى فقد قام OleDbDataAdapter بإحضار البيانات ومن تم وضعها في DataTable لينتهي بها الأمر في آخر المطاف على DataGridView.

إذا أردنا أن نظهر كل الحقول من غير حاجة إلى شريط التمرير Scrollbar فيجب أن نضيف السطر التالي من الكود في Form1_Load:

```
DGV.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
```

ليصبح مظهر الداتاغريدفيو كما يلي:



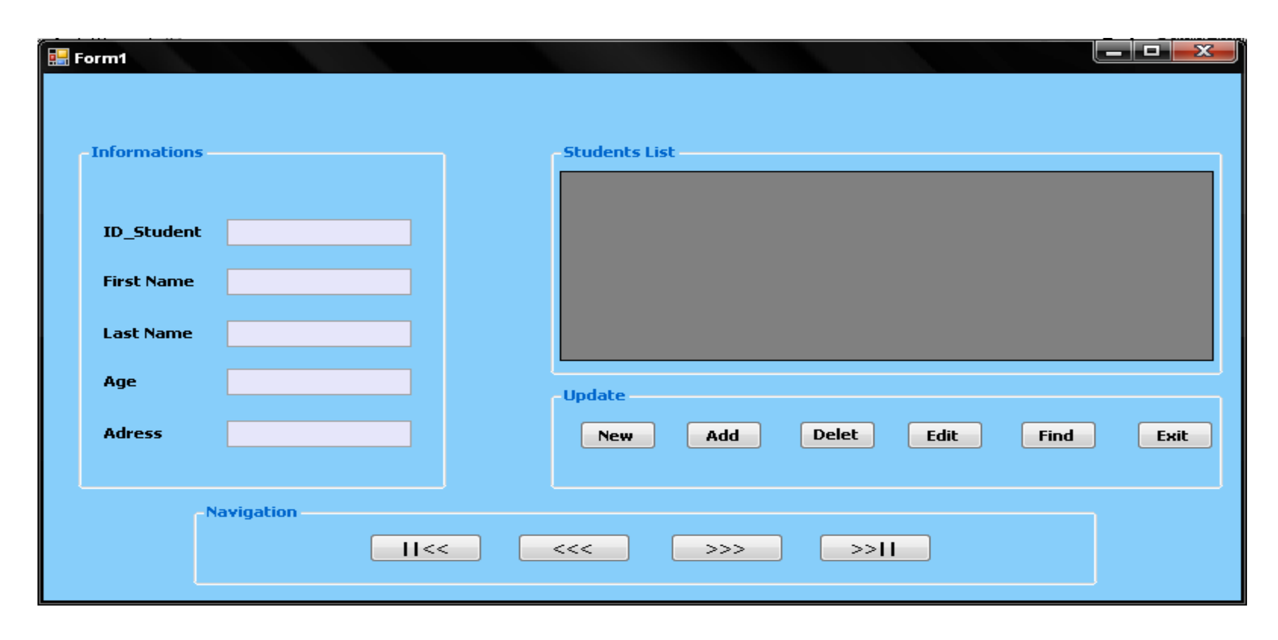
	ID_Student	First_Name	Last_Name	Age	Adress
▶	1	Khalid	ESSAADANI	22	FBS
	2	Hamid	MAKBOUL	22	Beni Mellal
	3	Mohammed	ELKHAL	23	Khouribga
*					



يمكننا القيام بنفس الأمر عن طريق نافذة الخصائص الخاصة ب DataGridView ثم نذهب إلى الخاصية AutoSizeColumnsMode و نضبطها على القيمة Fill.

أتوقع أن تصل إلى نفس النتيجة إذا اتبعت معي نفس المراحل، في حالة حدوث العكس أستجديك أن تعيد قراءة التطبيق حرفا حرفا.

الآن سنغير إن شاء الله مظهر الفورم ليصبح كما يلي، حتى نتمكن من تعلم أشياء أخرى:



الأدوات المستعملة في الفورم:

- أداة مجمع الأدوات GroupBox: وهي تلك الإطارات البيضاء الظاهرة في الفورم والتي تحيط بكل مجموعة على حدة.



- أداة مربع النص TextBox : وهي مربعات النص الموضوعة في جزء المعلومات Informations
- أداة الداتاغريديفيو DataGridView: وهي ذلك المستطيل الرمادي
- أداة الزر Button: وهي كل الأزرار الموجودة في أسفل الفورم

كما تلاحظ معي فهذا الفورم يتكون من أربعة أجزاء:

- Informations: من خلاله سنقوم بإظهار معلومات الطالب عند القيام بعملية البحث، وكذلك سنستعمله من أجل إضافة طالب جديد.
- Students List: وتقوم بعرض كل الطلبة الموجودين في قاعدة البيانات، وكذلك الطلبة الذين تتم إضافتهم.
- Update: وكما تلاحظ فهذا الجزء يضم مجموعة من الأزرار وهي كالتالي:
  - New: ويمكننا من تفريغ محتوى الجزء Informations.
  - Add: ويمكننا من إضافة طالب جديد.
  - Delete: ويمكننا من حذف طالب من جدول الطلاب.
  - Edit: ويمكننا من تعديل بيانات الطلاب.
  - Find: ويمكننا من البحث عن معلومات طالب معين عن طريق رقمه.
  - Exit: ويمكننا من الخروج من البرنامج.
- Navigation: ويمكننا من التنقل بين الطلاب بالاعتماد على أزرار التنقل.



سنورد شفرات كل هذه الأزرار الآن وسنرفق ذلك بشرح دقيق حتى ترسخ الشفرات في أذهاننا.

للإشارة فقط، سنستعمل إن شاء الله الوضعين معا (المتصل وغير المتصل) حتى نتمكن من إتقان الطريقتين معا.

1. الزر الأول New:

من خلال دوره، يتضح أن لا علاقة له ب ADO.Net فهو زر بسيط يقوم فقط بمسح محتوى الخانات النصية، وبالتالي فشفرته هي كالتالي:

```
For Each K As Control In GroupBox1.Controls
    If TypeOf K Is TextBox Then
        K.Text = ""
        TxtID.Focus()
    End If
Next
```

كما ترون فإن كل أدوات التيكست بوكس موجودة داخل أداة GroupBox1، ولذلك قمنا بتفريغها في دفعة واحدة عوض تفريغ كل تيكست بوكس في سطر !

أما فيما يخص السطر TxtID.Focus() فيقوم بوضع مؤشر الفأرة في أول تيكست بوكس .

2. الزر الثاني Add

حتى تتضح الرؤية إن شاء الله فإن الغاية من هذا الزر هو إضافة الطالب الجديد الذي تكتب بياناته في الفورم إلى جدول الطلاب في قاعدة البيانات.



سنرى كيف نقوم بذلك في الوضعين المتصل والمنفصل.

أولا : في الوضع المتصل On-line Mode:

كما رأينا فيما سبق، فالوضع المتصل يستعمل الكائن Command من أجل القيام بعمليات الإضافة والحذف والتعديل.

سنقوم بالذهاب إلى الجزء الخاص بالإعلان عن المتغيرات، وسنقوم بالإعلان عن كائن من نوع Command ولنسمه Cmd.

```
    Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source= StudentBase.accdb")
    Dim Da As OleDbDataAdapter
    Dim Dt As New DataTable
    Dim Cmd As New OleDbCommand
```

الآن بعد أن أعلنا عنه، سنتوجه إلى زر الإضافة Add ونضغط عليه مرتين ونكتب الشفرة التالية:

```
Try
            Cmd = New OleDbCommand("Insert Into Student Values
                                  ('" & Me.TxtID.Text & "','" _
                                     & Me.TxtFname.Text & "','" &
                                     Me.TxtLname.Text & "', '" _
                                  & Me.TextAge.Text & "','" &
                                  Me.TextAdress.Text & "')", Con)
            Con.Open()
            Cmd.ExecuteNonQuery()
```




```
            Con.Close()
        MsgBox("Student added successfully !",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```

قد يبدو السطر الخاص ب Command غريبا نوعا ما، نظرا لكثرة الرموز فيه ولكن لا عليك، سنشرحه الآن بإذن الله :

```
Cmd =New OleDbCommand("SQL Statement",ConnectionName)
```

الكائن OleDbCommand يأخذ بارامترين، أولهما نص الاستعلام المراد تنفيذه، والثاني اسم كائن الاتصال.

في حالتنا هذه، نص الاستعلام عبارة عن أمر الإضافة :

```
Insert Into Student Values ('"   & Me.TxtID.Text & "','" _
                                  & Me.TxtFname.Text & "','" &
                                 Me.TxtLname.Text & "', '" _
                                  & Me.TextAge.Text & "','" &
                                 Me.TextAdress.Text & "')
```

نضع القيم التي نريد إضافتها بين مزدوجتين ("")ونفصل بينها بفواصل(,)، أما الكوت (') فهو يستعمل فقط عندما يكون الحقل نصيا، ولكن تفاديا لوقوع أي أخطاء نقوم بكتابته في كل الحقول بما في ذلك



الحقول الرقمية، أما رمز الأندرسكور (_) فنستعمله للربط بين الأسطر، ويمكننا الاستغناء عنه إذا كتبنا كل الاستعلام في سطر واحد.

بإمكاننا وضع الاستعلام في متغير نصي من نوع String ثم نضع اسم هذا المتغير في البارامتر الأول من الكائن:

```
  Dim SQL As String = "Insert Into Student Values ('" &
                               Me.TxtID.Text & "','" _
                               & Me.TxtFname.Text & "','" &
                               Me.TxtLname.Text & "', '" _
                               & Me.TextAge.Text & "','" &
                               Me.TextAdress.Text & "')"

Cmd = New OleDbCommand(SQL, Con)
```

إذا تمكنت من استيعاب هذا السطر فالباقي أسهل بكثير، وإن وجدت أدنى صعوبة فراجعه جيدا وحاول أن تفهمه أكثر، وإن اقتضى الأمر جربه عمليا.

بعد كتابة أمر الاستعلام، وتحديد كائن الاتصال وفي حالتنا هذه هو Con، نقوم بفتح الاتصال عن طريق

```
Con.Open()
```

ثم ننفذ أمر Cmd:

```
Cmd.ExecuteNonQuery()
```



ثم نغلق الاتصال ونظهر رسالة مفادها أن الإضافة تمت بنجاح.

```
Con.Close()
MsgBox("Student added successfully !", MsgBoxStyle.Information)
```

ثم نقوم بتفريغ الكائن Dt حتى لا تتكرر البيانات القديمة، بعدها نقوم بالنداء على حدث تحميل الفورم Form1_Load الذي سيقوم بإعادة ملء Dt من جديد ثم يظهرها على الداتغريدڤيو.

```
Dt.Clear()
Form1_Load(sender, e)
```

عند حدوث أي خطأ ستظهر الرسالة الموجودة في الجزء الثاني من Try..Catch

```
Catch ex As Exception
            MsgBox("Some errors was occured !", MsgBoxStyle.Critical)
End Try
```

ثانيا : في الوضع المنفصل Off-line Mode:

لإضافة سطر جديد في الوضع المنفصل، نقوم بالإعلان عن كائن من نوع DataRow ، ثم نضع كل قيمة من قيم مربعات النص في خانة من خانات هذا الكائن، بعد ذلك نضيف هذا السطر الجديد إلى الكائن Dt، ثم نُفعِّل الإضافة في قاعدة البيانات عن طريق كائن جديد اسمه OleDbCommandBuilder.

أولا نعلن عن الكائن OleDbCommandBuilder ولنسمه Cmdb.



يأخذ هذا الكائن اسم الكائن OleDbDataAdapter كَبَرَامِتْر، ومن تم أي تعديل يقوم به هذا الأخير يتم تفعيله في قاعدة البيانات.

```
    Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source= StudentBase.accdb")
    Dim Da As OleDbDataAdapter
    Dim Dt As New DataTable
    Dim Cmd As New OleDbCommand
    Dim Dr As OleDbDataReader
    Dim Cmdb As New OleDbCommandBuilder
```

بعد الإعلان عنه نقوم بكتابة شفرة الإضافة في الوضع المنفصل:

```
        Try
            Dim R As DataRow = Dt.NewRow
            R(0) = TxtID.Text
            R(1) = TxtFname.Text
            R(2) = TxtLname.Text
            R(3) = TextAge.Text
            R(4) = TextAdress.Text
            Dt.Rows.Add(R)
            Cmdb = New OleDbCommandBuilder(Da)
            Da.Update(Dt)
            MsgBox("Student added successfully !",
MsgBoxStyle.Information)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```



تأمل جيدا طريقة إنشاء سطر فارغ في Dt وكيفية ملئه عن طريق مربعات النص (القيم المراد إضافتها)، وكيف قمنا بإضافة السطر المنشأ إلى Dt، و في الأخير كيف فعلنا التحديث بواسطة السطرين، وهذان السطران هما بيت القصيد في شفرة الإضافة في الوضع المنفصل:

```
        Try
            Dim R As DataRow = Dt.NewRow
            R(0) = TxtID.Text
            R(1) = TxtFname.Text
            R(2) = TxtLname.Text
            R(3) = TextAge.Text
            R(4) = TextAdress.Text
            Dt.Rows.Add(R)
            Cmdb = New OleDbCommandBuilder(Da)
           Da.Update(Dt)
            MsgBox("Student added successfully !",
MsgBoxStyle.Information)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```

أتمنى أن تكون الشروح مفهومة، إذا استعصى عليك أمر معين فاسأل الله أن يعينك وسامحني على تقصيري وابذل مجهودا وإن شاء الله ستفهمه بِيُسْر، لأن هذه الشفرات سهلة جدا وبالممارسة ستغدو سائغة كشرب الماء فيما يلي سنسرد الشفرتين معا، حتى يتأتى لك مقارنتهما واستيعابهما، ولكن احذر أن تنفذهما مرة واحدة، فللشفرتين نفس الدور ☺:




```
    Private Sub Add_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Add.Click

        'On-line Mode
        Try
            Cmd = New OleDbCommand("Insert Into Student Values
                             ('" & Me.TxtID.Text & "','" _
                                  & Me.TxtFname.Text & "','" &
                                  Me.TxtLname.Text & "', '" _
                                 & Me.TextAge.Text & "','" &
                                 Me.TextAdress.Text & "')",
                                 Con)
            Con.Open()
            Cmd.ExecuteNonQuery()
            Con.Close()
            MsgBox("Student added successfully !",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try




        'Off-line Mode
        Try
            Dim R As DataRow = Dt.NewRow
            R(0) = TxtID.Text
            R(1) = TxtFname.Text
            R(2) = TxtLname.Text
            R(3) = TextAge.Text
            R(4) = TextAdress.Text
            Dt.Rows.Add(R)
```




```
                Cmdb = New OleDbCommandBuilder(Da)
                Da.Update(Dt)
                MsgBox("Student added successfully !", 
MsgBoxStyle.Information)
            Catch ex As Exception
                MsgBox("Some errors was occured !", 
MsgBoxStyle.Critical)
            End Try
        End Sub
```



3. زر الحذف Delete

عند الضغط على هذا الزر يظهر مدخل للنص، يطلب من المستخدم كتابة رقم الطالب المراد حذفه، بعد كتابة رقم الطالب والضغط على OK يتم حذف الطالب مباشرة.

في حالة حذف الطالب تظهر رسالة تعلم المستخدم بنجاح العملية، وفي حالة العكس تظهر رسالة تنبئ المستخدم بذلك.

أولا : في الوضع المتصل On-line Mode:

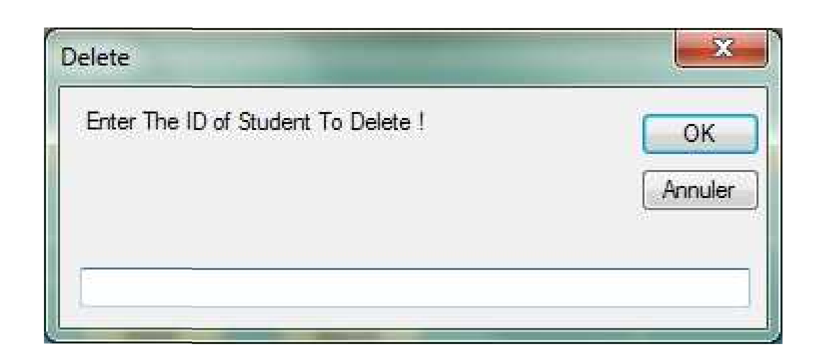
```
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To
Delete !", "Delete")
            Cmd = New OleDbCommand("Delete * From Student
where ID_Student=" & Input & "", Con)
            Con.Open()
            Cmd.ExecuteNonQuery()
            Con.Close()
            MsgBox("Student Deleted Successfully",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```

قمنا بالإعلان عن متغير نصي أسميناه: Input

```
Dim Input As String
```



ثم حفظنا فيه القيمة المدخلة بواسطة المستخدم، بالمناسبة ستظهر للمستخدم النافذة التالية:




```
Input = InputBox("Enter The ID of Student To Delete !",
"Delete")
```

ثم نقوم بتنفيذ استعلام حذف الطالب الذي يتوافق رقمه مع القيمة المدخلة من طرف المستخدم.

```
        Cmd = New OleDbCommand("Delete * From Student where
ID_Student=" & Input & "", Con)
        Con.Open()
        Cmd.ExecuteNonQuery()
        Con.Close()
```

في الأخير نقوم بإظهار رسالة تفيد بحذف الطالب، ثم نفرغ Dt لنملأها بالبيانات الجديدة عن طريق النداء على حدث تحميل الفورم.

```
            MsgBox("Student Deleted Successfully",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
```



ثانيا : في الوضع المنفصل Off-line Mode :

```
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To
Delete !", "Delete")
            For i As Integer = 0 To Dt.Rows.Count - 1
                If Input = Dt.Rows(i).Item(0) Then
                    Dt.Rows(i).Delete()
                    Cmdb = New OleDbCommandBuilder(Da)
                    Da.Update(Dt)
                    MsgBox("Student Deleted Successfully",
MsgBoxStyle.Information)
                    Exit Sub
                End If
            Next
            MsgBox("Not Found !")
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```

نفس الخطوات السابقة، بعد ذلك نقوم بالبحث داخل Dt عن الطالب الذي يحمل الرقم المدخل، وذلك عن طريق الذهاب من أول عنصر في Dt إلى آخر عنصر، وفي كل مرة نقوم بمقارنة بين قيمة المتغير Input وبين رقم الطالب الذي نصل إليه.

إذا تحقق شرط التوافق بين المتغير Input وبين رقم الطالب المبحوث عنه، نقوم بحذفه ابتدائيا من Dt ثم نفعل الحذف في قاعدة البيانات عن طريق Cmdb، بعد ذلك نعيد تحميل الفورم لتحديث محتوى الداتاغريدفيو.



والآن سنعرض شفرتي الحذف في كلا الوضعين حتى تتضح الرؤية :

```
Private Sub Delete_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Delete.Click
        'On-line Mode
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To
Delete !", "Delete")
            Cmd = New OleDbCommand("Delete * From Student
where ID_Student=" & Input & "", Con)
            Con.Open()
            Cmd.ExecuteNonQuery()
            Con.Close()
            MsgBox("Student Deleted Successfully",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try


        'Off-line Mode
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To
Delete !", "Delete")
            For i As Integer = 0 To Dt.Rows.Count - 1
                If Input = Dt.Rows(i).Item(0) Then
                    Dt.Rows(i).Delete()
                    Cmdb = New OleDbCommandBuilder(Da)
                    Da.Update(Dt)
                    MsgBox("Student Deleted Successfully",
MsgBoxStyle.Information)
                    Exit Sub
                End If
            Next
```




```
            MsgBox("Not Found !")
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```

تذكير: لا تجرب الشفرتين مرة واحدة.



4. زر البحث Find:

أما دور هدا الزر فهو البحث عن طالب معين بالاعتماد على رقمه، وعند العثور عليه تظهر بياناته في خانات النص.

أولا : في الوضع المتصل On-line Mode:

```
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To Search !", "Search")
            Cmd = New OleDbCommand("select * from Student where ID_Student=" & Input & "", Con)
            Con.Open()
            Dr = Cmd.ExecuteReader
            While Dr.Read()
                Me.TxtID.Text = Dr(0)
                Me.TxtLname.Text = Dr(1)
                Me.TxtFname.Text = Dr(2)
                Me.TextAge.Text = Dr(3)
                Me.TextAdress.Text = Dr(4)
            End While
            Dr.Close()
            Con.Close()
        Catch ex As Exception
            MsgBox("Some errors was occured !", MsgBoxStyle.Critical)
        End Try
```

في بداية حديثنا عن كائنات الوضع المتصل، كنا قد ذكرنا كائنا اسمه DataReader وقلنا بأن دوره هو قراءة البيانات التي يقوم Command بجلبها، كما يتيح لنا إمكانية استغلال القيم المقروءة لنظهرها في عملية البحث.



يتوفر هذا الكائن على مجموعة من الدوال لن نتطرق إلى ذكرها هنا، تفاديا للخروج عن سياق السلسلة، ولكن سنقوم ـ إن شاء الله ـ إذا أتيحت لنا فرصة أخرى ببسط كل هذه الكائنات وغيرها في كتاب آخر.

عند نهاية قراءة البيانات نقوم بإغلاق DataReader بنفس الطريقة التي نغلق بها الاتصال.

ثانيا : في الوضع المنفصل Off-line Mode :

عملية البحث في الوضع المنفصل شبيهة إلى حد بعيد بعملية الحذف في ذات الوضع، غير أننا عوض أن نقوم بحذف الطالب المعثور عليه نقوم بإظهار بياناته.

```
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To 
Search !", "Search")
            For i As Integer = 0 To Dt.Rows.Count - 1
                If Input = Dt.Rows(i).Item(0) Then
                    Me.TxtID.Text = Dt.Rows(i).Item(0)
                    Me.TxtLname.Text = Dt.Rows(i).Item(1)
                    Me.TxtFname.Text = Dt.Rows(i).Item(2)
                    Me.TextAge.Text = Dt.Rows(i).Item(3)
                    Me.TextAdress.Text = Dt.Rows(i).Item(4)
                    Exit Sub
                End If
            Next
            MsgBox("Not Found !")
        Catch ex As Exception
            MsgBox("Some errors was occured !", 
MsgBoxStyle.Critical)
        End Try
```



اقرأ الشفرة بتمعن، وستهضمها في لمح البصر... إن استعصيتها عليك بهذا الدعاء الجميل الذي عليك بملازمته كلما استصعبت أمرا ما: " اللهم لا سهل إلا ما جعلته سهلا، وأنت تجعل الحزن إذا شئت سهلا "

فيما يلي سنورد كلتا الشفرتين:

```
Private Sub Search_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Search.Click
        'On-line Mode
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To
Search !", "Search")
        Cmd = New OleDbCommand("select * from Student where
ID_Student=" & Input & "", Con)
            Con.Open()
            Dr = Cmd.ExecuteReader
            While Dr.Read()
                Me.TxtID.Text = Dr(0)
                Me.TxtLname.Text = Dr(1)
                Me.TxtFname.Text = Dr(2)
                Me.TextAge.Text = Dr(3)
                Me.TextAdress.Text = Dr(4)
            End While
            Dr.Close()
            Con.Close()
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try

        'Off-line Mode
        Dim Input As String
        Try
            Input = InputBox("Enter The ID of Student To
Search !", "Search")
```




```
            For i As Integer = 0 To Dt.Rows.Count - 1
                If Input = Dt.Rows(i).Item(0) Then
                    Me.TxtID.Text = Dt.Rows(i).Item(0)
                    Me.TxtLname.Text = Dt.Rows(i).Item(1)
                    Me.TxtFname.Text = Dt.Rows(i).Item(2)
                    Me.TextAge.Text = Dt.Rows(i).Item(3)
                    Me.TextAdress.Text = Dt.Rows(i).Item(4)
                    Exit Sub
                End If
            Next
            MsgBox("Not Found !")
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
    End Sub
```



5. زر التعديل Edit:

قبل التعديل، ينبغي أن نقوم أولا بعملية البحث عن الطالب المراد تغيير بياناته، بعد العثور عليه نعدل البيانات ثم نضغط على هذا الزر (ركز جيدا في هذه الخطوات)

بحث عن الطالب ← تغيير البيانات ← الضغط على زر التعديل لحفظ البيانات.

أولا : في الوضع المتصل On-line Mode:

```
        Try
            Cmd = New OleDbCommand("Update Student set
                                    First_Name='" &
                                    Me.TxtFname.Text & _
                                    "', Last_Name='" &
                                    Me.TxtLname.Text & _
                                    "' , Age='" &
                                    TextAge.Text & _
                                    "' ,Adress='" &
                                    Me.TextAdress.Text & _
                                    "' Where ID_Student=" &
                                    TxtID.Text & "", Con)
            Con.Open()
            Cmd.ExecuteNonQuery()
            Con.Close()
            MsgBox("Student Edited successfully !",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```



مثلها مثل شفرة الإضافة، نقوم فقط بتغيير الاستعلام من Insert إلى Update، لا تنبهر من مظهر تلك الرموز فهي فقط لتنظيم تسلسل الحقول، ضع في بالك فقط أن كل رمز يُبدأ به أولا يُختم به أخيرا !

ثانيا : في الوضع المنفصل Off-line Mode :

```
        Try
            For i = 0 To Dt.Rows.Count - 1
                If Dt.Rows(i).Item(0) = TxtID.Text Then
                    Dt.Rows(i).Item(0) = TxtID.Text
                    Dt.Rows(i).Item(1) = TxtFname.Text
                    Dt.Rows(i).Item(2) = TxtLname.Text
                    Dt.Rows(i).Item(3) = TextAge.Text
                    Dt.Rows(i).Item(4) = TextAdress.Text
                    Cmdb = New OleDbCommandBuilder(Da)
                    Da.Update(Dt)
                    MsgBox("Student Edited Successfully",
MsgBoxStyle.Information)
                End If
            Next
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
```

نقوم بالبحث في Dt عن الطالب الذي رقمه يساوي الرقم المكتوب في TxtID أي رقم الطالب المراد تعديله، بعد العثور عليه نقوم بإبدال القيم القديمة المخزنة في Dt بالقيم الجديدة المكتوبة في خانات النص، ثم نفعل التعديل عن طريق CommandBuilder.



فيما يلي سنورد كلتا الشفرتين:

```
Private Sub Edit_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Edit.Click
        'On-line Mode
        Try
            Cmd = New OleDbCommand("Update Student set
                                    First_Name='" &
                                    Me.TxtFname.Text & _
                                    "', Last_Name='" &
                                    Me.TxtLname.Text & _
                                     "' , Age='" &
                                    TextAge.Text & _
                                    "' ,Adress='" &
                                    Me.TextAdress.Text & _
                                     "' Where ID_Student=" &
                                    TxtID.Text & "", Con)
            Con.Open()
            Cmd.ExecuteNonQuery()
            Con.Close()
            MsgBox("Student Edited successfully !",
MsgBoxStyle.Information)
            Dt.Clear()
            Form1_Load(sender, e)
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try


        'Off-line Mode
        Try
            For i = 0 To Dt.Rows.Count - 1
                If Dt.Rows(i).Item(0) = TxtID.Text Then
                    Dt.Rows(i).Item(0) = TxtID.Text
                    Dt.Rows(i).Item(1) = TxtFname.Text
                    Dt.Rows(i).Item(2) = TxtLname.Text
                    Dt.Rows(i).Item(3) = TextAge.Text
                    Dt.Rows(i).Item(4) = TextAdress.Text
```




```
                    Cmdb = New OleDbCommandBuilder(Da)
                    Da.Update(Dt)
                    MsgBox("Student Edited Successfully",
MsgBoxStyle.Information)
                End If
            Next
        Catch ex As Exception
            MsgBox("Some errors was occured !",
MsgBoxStyle.Critical)
        End Try
    End Sub
```



6. زر الخروج Exit:

قم بالضغط عليه مرتين واكتب الكلمة التالية:

```
Private Sub Exit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Exit.Click

        End

    End Sub
```

الآن انتهينا ولله الحمد من العمليات الرئيسية من إضافة و حذف وبحث وتعديل، بقي لنا الآن إن شاء الله القيام بعملية التنقل بين سجلات الطلاب، وهذه العملية من أسهل ما يكون، سنقوم بإنشاء إجراء Procedure نسميه ()MoveData يتلقى رتبة السجل كبارامتر MoveData (Position as Integer)، ثم يذهب إلى تلك الرتبة ويظهر قيمها في خانات النص.

نود الإشارة فقط إلى أن أول سطر في Dt رتبته هي :0

وأن السطر الأخير رتبته هي: Dt.Rows.Count-1

أما رتبة السطر التالي فهي تزايدية بمقدار واحد: Position=Position+1

وبالمقابل رتبة السطر السابق تكون تناقصية بمقدار واحد: Position=Position-1



سيكون نص شفرة الإجراء MoveData كما يلي :

```
Sub MoveData(ByVal Position As Integer)
    TxtID.Text = Dt.Rows(Position).Item(0)
    TxtFname.Text = Dt.Rows(Position).Item(1)
    TxtLname.Text = Dt.Rows(Position).Item(2)
    TextAge.Text = Dt.Rows(Position).Item(3)
    TextAdress.Text = Dt.Rows(Position).Item(4)
End Sub
```

قبل النداء على هذا الإجراء في أزرار التنقل، نقوم بالإعلان عن متغير رقمي عام، ولنسمه مثلا: Index

```
Imports System.Data.OleDb
Public Class Form1
    Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=Students.accdb")
    Dim Da As OleDbDataAdapter
    Dim Dt As New DataTable
    Dim Cmd As New OleDbCommand
    Dim Dr As OleDbDataReader
    Dim Cmdb As New OleDbCommandBuilder
   Dim Index As Integer
```


الآن نطبق ما تحدثنا عنه قبل قليل، نذهب إلى زر "الأول" ونقوم بالنداء على هذا الإجراء مع تغيير قيمة الرتبة ب 0:

```
Private Sub First_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles First.Click

        Index = 0
        MoveData(Index)
    End Sub
```


و في الزر "الأخير" :




```
Private Sub Last_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Last.Click
        Index = Dt.Rows.Count - 1
        MoveData(Index)
 End Sub
```

شفرة الزر السابق :

```
Private Sub Previous_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Previous.Click

        If Index = 0 Then
            MsgBox("This is the first student",
MsgBoxStyle.Critical)
            Exit Sub
        End If
        Index = Index - 1
        MoveData(Index)

    End Sub
```

نتحقق من الرتبة في كل مرة حتى لا يقع خطأ عند الضغط المتواصل على هذا الزر.

شفرة الزر التالي :

```
Private Sub Next_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Next.Click

        If Index = Dt.Rows.Count - 1 Then
            MsgBox("This is the Last student", MsgBoxStyle.Critical)
            Exit Sub
        End If
        Index = Index + 1
        MoveData(Index)
    End Sub
```



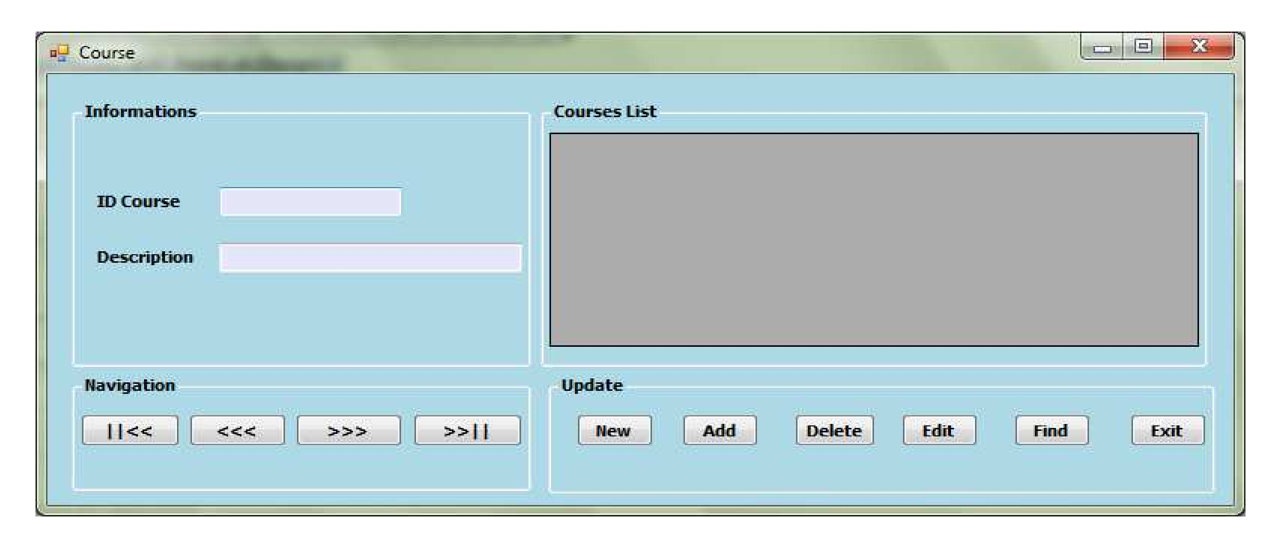
تمرين تحصيلي :

حتى نقوم بتحصيل و فهم كل الشفرات التي أوردناها آنفا، سنسعى إلى إنجاز هذا التمرين المماثل (أو بالأحرى ستسعى لذلك وحدك، كان الله في عونك ☺)

المطلوب منك عزيزي القارئ أن تنجز الفورم التالي الذي يقوم بإدارة المواد، بحيث من خلاله يستطيع المستخدم أن يشاهد كل المواد المحفوظة في قاعدة البيانات، وكذا إمكانية التعديل عليها وحذفها و إضافة مواد جديدة، زيادة على أزرار التنقل بين سجلات المواد.

الأمر بسيط جدا، كل ما عليك هو مراجعة الشفرات السابقة و محاولة استيعابها أكثر، و إن شاء الله ستنجزه في وقت يسير !

هنا صورة الفورم المطلوب منك إنجازه :



لك الحرية في استخدام أي وضع تشاء (المنفصل أو المتصل) !



الجزء الثالث:

مجموعة من الشفرات للحاجة



بصراحة، أردت أن أقدم فيما يلي شرح كيفية إنجاز فورم إضافة النقاط للطلاب في المواد التدريسية، ولكنني ارتأيت إلغاء ذلك لأن الكتاب سيطول كثيرا، ولأن التحليل أصلا ليس كاملا، إذ أن برنامج إدارة الطلاب يتطلب تحليلا قبل بداية الإنجاز، أما في سلسلتنا هذه فنحن نركز فقط على ما هو عملي فحسب، إذ بإتقانك لهذا التطبيق ستغدو قادرا إن شاء الله على إنجاز غيره مهما كان نص المشروع.

كما قلت، ما يهم في سلسلتنا هذه هو التمكن من إنجاز الأعمال باحترافية، ولا يهم المثال الذي نعطيه، لأن الغرض ليس هو المثال نفسه ولكن الغرض هو تعلم تقنيات إنجازه.

غير أن هذا لن يمنعنا من إيراد بعض الأكواد التي قد تحتاجها إن شاء الله في بعض التطبيقات المرتبطة بقاعدة بيانات.

الشفرة الأولى : كيفية ملأ كومبوبوكس Combobox من جدول بقاعدة البيانات:

```
Imports System.Data.OleDb

Public Class Marks
    Dim Con As New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=Students.accdb")
    Dim Da As OleDbDataAdapter
    Dim Dt As New DataTable

    Private Sub Combo_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        Da = New OleDbDataAdapter("Select
ID_Student,Last_Name + ' '+ First_Name as FullName From
student", Con)
        Da.Fill(Dt)
        Me.ComboBox1.DataSource = Dt
        Me.ComboBox1.DisplayMember = "FullName"
```




```
            Me.ComboBox1.ValueMember = "ID_Student"
        End Sub
    End Class
```

نقوم بجلب البيانات المراد إظهارها في الكومبوبوكس، إضافة إلى الحقل الرئيسي (ليس ضروريا إذا كان الغرض هو الإظهار فقط)، بعد ذلك نقوم بوضع البيانات المجلوبة من طرف DataAdapter ونضعها في DataTable.

بعد ملء DataTable، نجعلها مصدر بيانات للكومبوبوكس.

```
        Me.ComboBox1.DataSource = Dt
```

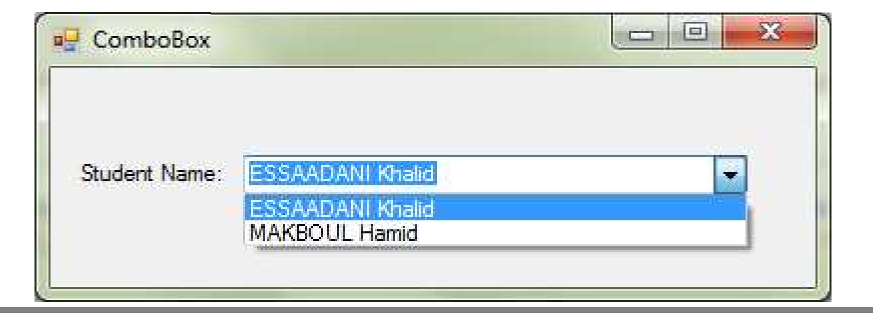
ثم نحدد الحقل المراد إظهاره بالاعتماد على الخاصية DisplayMember، في حالتنا هذه سنظهر الاسم الشخصي والعائلي معا، ولعلك لاحظت ذلك في الاستعلام الذي قمنا به، إذ قمنا بدمجهما معا.

```
        Me.ComboBox1.DisplayMember = "FullName"
```

و في الختام نحدد الحقل الرئيسي للجدول الذي جلبنا منه البيانات.

```
        Me.ComboBox1.ValueMember = "ID_Student"
```

و هنا صورة للنتيجة التي وصلنا إليها :





الشفرة الثانية: كيفية إظهار بيانات معينة عند اختيار عنصر من الكومبوبوكس Combobox

الحدث الذي يتولد عندما نقوم باختيار عنصر من القائمة المنسدلة ComboBox يسمى SelectedIndexChanged، وبالتالي فشفرتنا هذه يجب أن نضعها ضمن هذا الحدث، ولتفادي وقوع أي خطأ أثناء التنفيذ نضيف الأمر Try...Catch :

```
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged

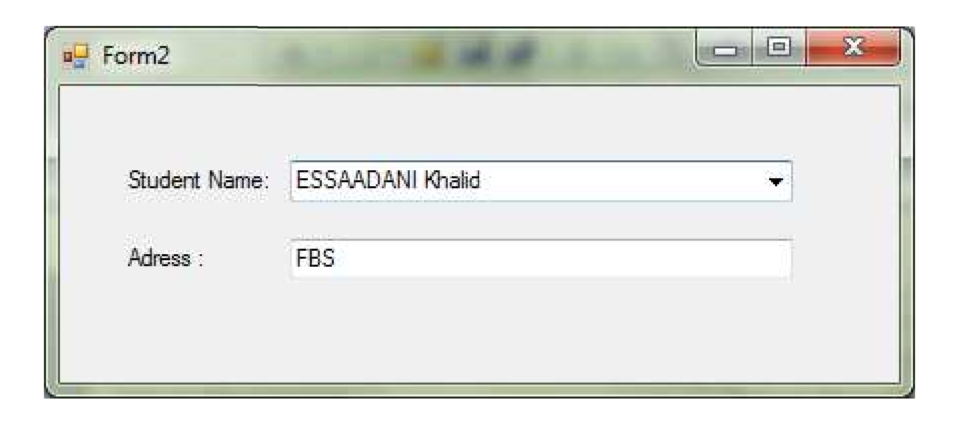
        Try
            Dt2.Clear()
            Da = New OleDbDataAdapter("select Adress From
Student Where ID_Student=" & ComboBox1.SelectedValue & "",
Con)
            Da.Fill(Dt2)
            Me.TextBox1.Text = Dt2.Rows(0).Item(0)
        Catch ex As Exception
            Exit Sub
        End Try

    End Sub
```

فقط للإشارة لا يلزمنا استعمال DataTable واحد، لأنه لا يعقل أن يكون هو مصدر البيانات وفي نفس الوقت هو النتيجة، لذا وجب الإعلان عن متغير آخر ولنسمه مثلا Dt2 كما في الأنموذج أعلاه.

أول سطر يقوم بتفريغ Dt2 عند كل اختيار، من أجل ملئه بالبيانات الجديدة، بعد ذلك نملؤه بالطرق التي رأيناها آنفا، ثم نظهر الحقل الذي نريد، بعد التنفيذ ستكون النتيجة كما يلي :





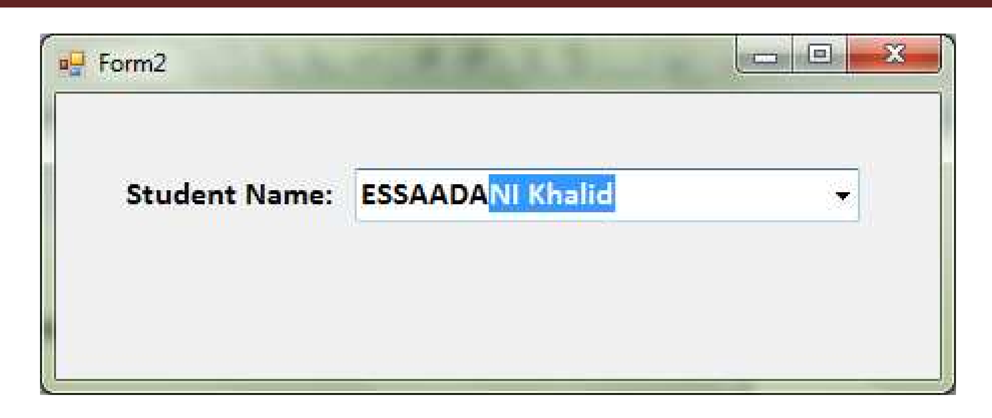
عند اختيار اسم الطالب، سيظهر عنوانه في خانة النص السفلية.

الشفرة الثالثة: كيفية إضافة الكتابة الذاتية IntelliSense عند الكتابة في Combobox ، كما هو الحال في متصفح الويب غوغل كروم Google Chrome

لعلك لاحظت أن العديد من البرامج والمواقع تستعمل هذه التقنية في البحث، إذ بمجرد ما تبدأ كتابة الأحرف الأولى للبحث، يقوم البرنامج تلقائيا بإكمال الأحرف المتبقية.

الشفرة التي سنعرضها بعد قليل، تقوم بإنجاز هذه المهمة، وهذه صورة على النتيجة المرتقبة :





الحدث الذي يتولد عند الضغط على زر في لوحة المفاتيح هو KeyUp، وفي حالتنا هذه هو الأنسب، لأنه عند الضغط على كل زر نقوم بعملية البحث داخل الكومبوبوكس والمقارنة، ومن تم إكمال الحروف المكتوبة.

```
Private Sub ComboBox1_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles ComboBox1.KeyUp

        Dim index As Integer
        Dim found As String
        Dim actual As String
        Dim bo As Boolean = e.KeyCode = Keys.Back Or e.KeyCode
= Keys.Left _
        Or e.KeyCode = Keys.Right Or e.KeyCode = Keys.Up Or
e.KeyCode = Keys.Down _
        Or e.KeyCode = Keys.Delete Or e.KeyCode =
Keys.PageDown _
        Or e.KeyCode = Keys.PageUp Or e.KeyCode = Keys.End _
        Or e.KeyCode = Keys.Home

        If bo = True Then
            Exit Sub
        End If

        actual = Me.ComboBox1.Text
        index = Me.ComboBox1.FindString(actual)
```




```
        If index > -1 Then
            found = Me.ComboBox1.Items(index).ToString
            Me.ComboBox1.SelectedIndex = index
            Me.ComboBox1.SelectionStart = actual.Length
            Me.ComboBox1.SelectionLength = found.Length
        End If
    End Sub
```

لا تنزعج من هذه الشفرة، لأنها هكذا تبدو عند أول وهلة، ركز معي في الشرح وستكتشف بأن الأمر يسير جدا 🙂:

الأسطر الأولى :

```
        Dim index As Integer
        Dim found As String
        Dim actual As String
```

المتغير index من نوع رقمي، دوره هو البحث عن رتبة العنصر الذي يشبه في بدايته النص المكتوب، بمعنى آخر، لو كان النص المكتوب هو KARIM فإنه سيذهب ويبحث داخل عناصر الكومبوبوكس عن رتبة العنصر الذي يبدأ بنفس الأحرف KARIM.

بعد العثور على رتبة العنصر المطابق مبدئيا للنص المكتوب، نقوم بتخزين قيمته في المتغير found.

حتى لا تختلط عليك المفاهيم، فرتبة العنصر هي رقمه الترتيبي داخل القائمة، أما قيمته فهي العنصر ذاته أي نص العنصر.



أما المتغير النصي actual فهو يساوي قيمة النص المكتوب في الكومبوبوكس.

```
Dim bo As Boolean = e.KeyCode = Keys.Back _
                 Or e.KeyCode = Keys.Left _
                 Or e.KeyCode = Keys.Right _
                 Or e.KeyCode = Keys.Up _
                 Or e.KeyCode = Keys.Down _
                 Or e.KeyCode = Keys.Delete _
                 Or e.KeyCode = Keys.PageDown _
                 Or e.KeyCode = Keys.PageUp _
                 Or e.KeyCode = Keys.End _
                 Or e.KeyCode = Keys.Home
```

المتغير bo نقوم فيه بحفظ حالة الأزرار الواردة أعلاه وهي: زر الحذف والرجوع إلى الخلف و أزرار الأسهم وغيرها من الأزرار التي ينبغي أن تُستثنى من عملية الكتابة.

وفيما يلي دورها الرئيسي :

```
If bo = True Then
    Exit Sub
End If
```

نتأكد من الزر الذي ضغط عليه المستخدم، فإن كان يساوي أحد الأزرار المحفوظة في المتغير bo نقوم بإلغاء باقي الكود، أما إذا كان الزر المضغوط خلاف ذلك، فسينفذ باقي الكود.

```
actual = Me.ComboBox1.Text
index = Me.ComboBox1.FindString(actual)
```



السطر الأول يقوم بتخزين النص المكتوب في المتغير actual، أما السطر الثاني فيقوم بالبحث عن رتبة العنصر الذي يشبه في بداياته النص المكتوب، وذلك عن طريق الداله FindString

```
If index > -1 Then
    found = Me.ComboBox1.Items(index).ToString
    Me.ComboBox1.SelectedIndex = index
    Me.ComboBox1.SelectionStart = actual.Length
    Me.ComboBox1.SelectionLength = found.Length
End If
```

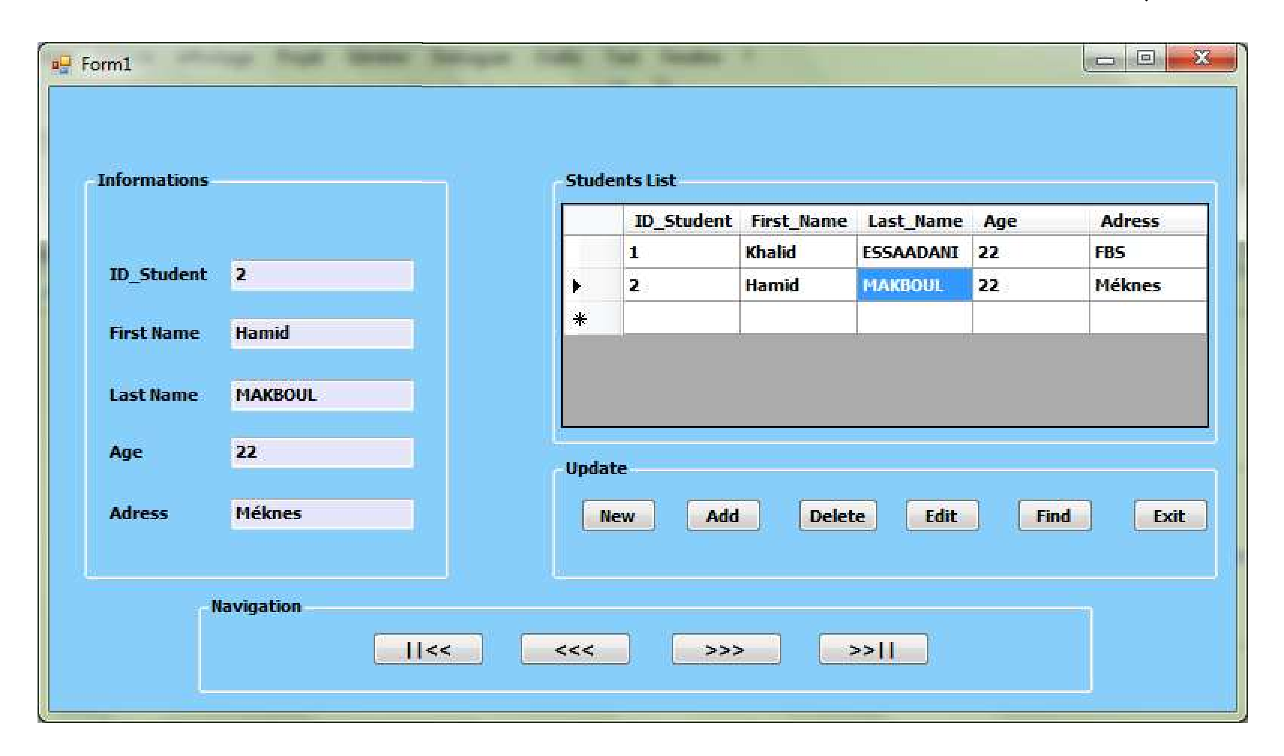
نتحقق من قيمة المتغير index فإن كانت أصغر من 1- فهذا يعني بأنه لا يوجد أي عنصر له بداية مشابهة للنص المكتوب، أما إن كانت قيمته أكبر من 1- فهذا يدل على أنه تم العثور على عنصر يبتدئ بالنص المكتوب.

بعد تحقق الشرط، أي العثور على عنصر يبدأ بالنص المكتوب داخل الكومبوبوكس ، نقوم بحفظ قيمة العنصر المعثور عليه في المتغير found، ثم نغير العنصر الافتراضي بالعنصر المعثور عليه ونظهره في الكومبوبوكس، بعد ذلك نقوم بتحديد تتمة النص المكتوب عن طريق الدالة SelectionStart، وفي الأخير نحدد طول التحديدSelectionLength وهو يبدأ من نهاية النص المكتوب إلى نهاية العنصر المعثور عليه.

كان الله في عونك أرهقتك ببعض الكلمات...ولكن والحق يقال، لم أجد أيسر من ذلك للشرح

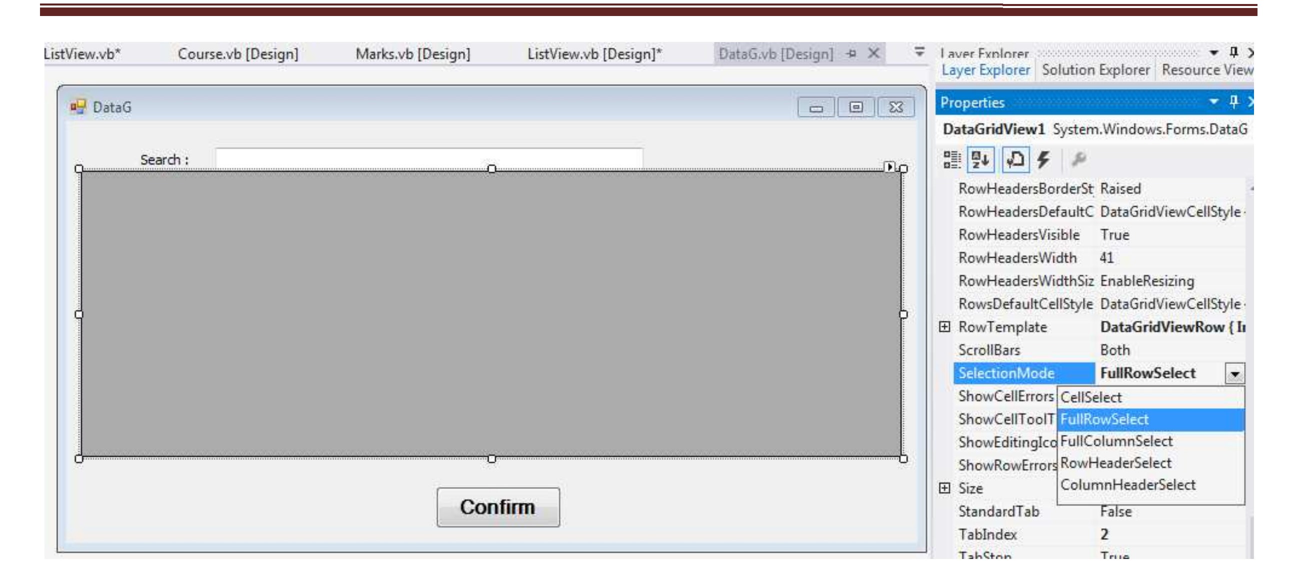


الشفرة الرابعة: كيفية إظهار بيانات سطر من الداتاغريدڤيو على خانات النص بمجرد تحديده

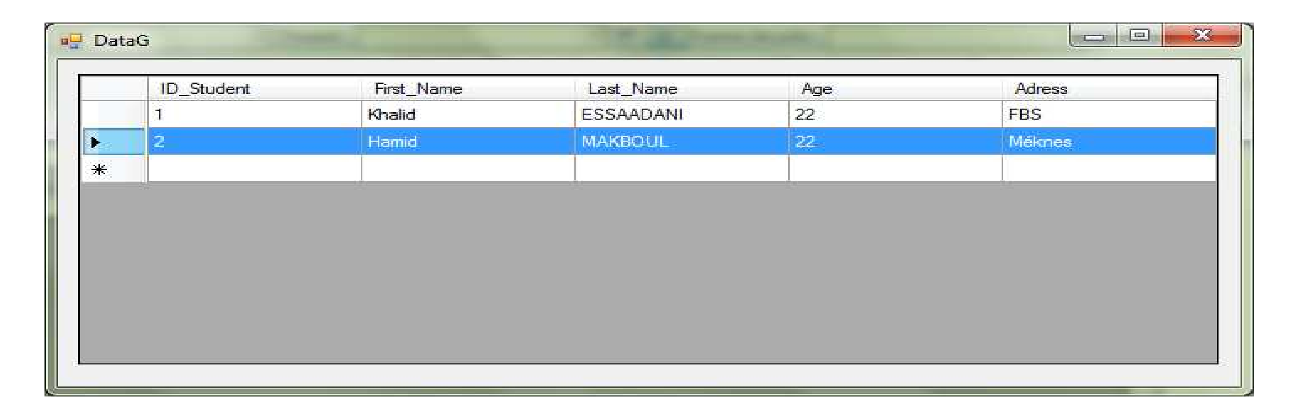


عند تحديد سطر معين من أداة الداتاغريدڤيو، فإن التحديد يشمل فقط الخانة المضغوط عليها، ولجعل التحديد يشمل كل السطر قم بالذهاب إلى نافذة الخصائص وغير قيمة الخصيصة SelectionMode من القيمة الافتراضية إلى القيمة FullRowSelect كما توضح الصورة أسفله:





هذه المرة سيصبح تحديد السطر كاملا :



ID_Student	First_Name	Last_Name	Age	Adress
1	Khalid	ESSAADANI	22	FBS
2	Hamid	MAKBOUL	22	Méknes

نعود إلى السؤال، المطلوب منا هذه المرة هو تحديد بيانات السطر المحدد على خانات النص عند الضغط عليه، الحدث المناسب لهذه الحالة هو Click الخاص بالداتاغريدڤيو، ثم نكتب الشفرة التالية:




```
Private Sub DGV_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles DGV.Click

        Try

            Dim Pos As Integer = Me.DGV.CurrentRow.Index

            Me.TxtID.Text = DGV.Rows(Pos).Cells(0).Value
            Me.TxtFname.Text = DGV.Rows(Pos).Cells(1).Value
            Me.TxtLname.Text = DGV.Rows(Pos).Cells(2).Value
            Me.TextAge.Text = DGV.Rows(Pos).Cells(3).Value
            Me.TextAdress.Text = DGV.Rows(Pos).Cells(4).Value
        Catch ex As Exception
            Exit Sub
        End Try

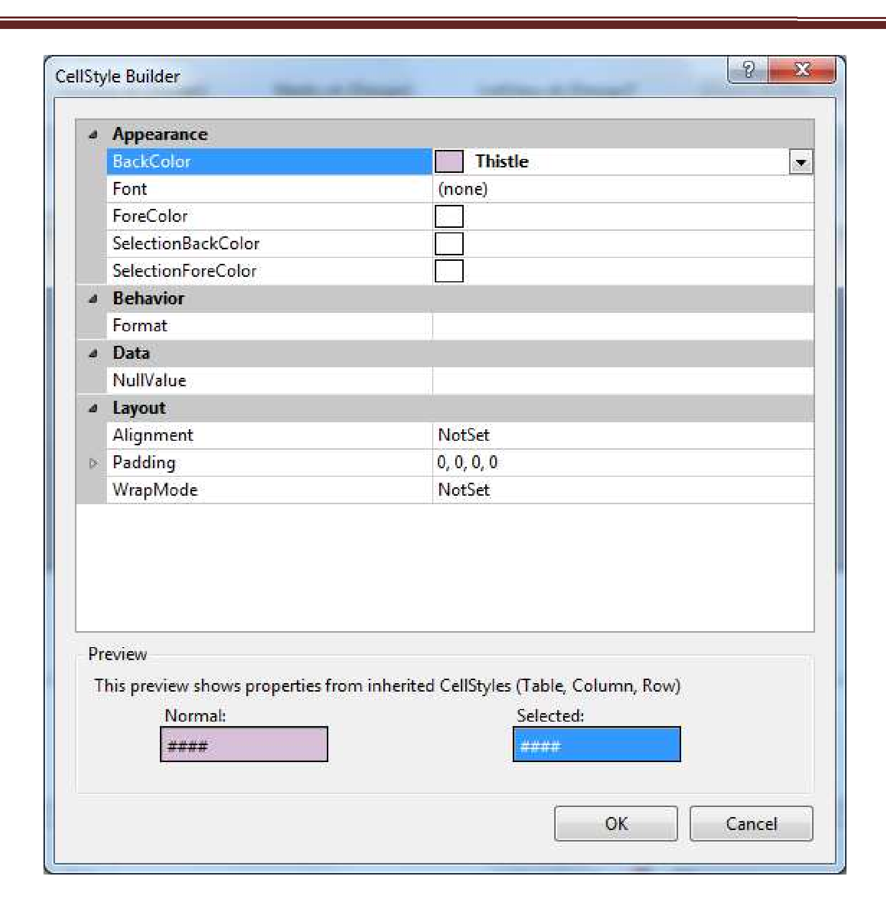
    End Sub
```

قمنا بالإعلان عن متغير رقمي أسميناه Pos من أجل حفظ رتبة السطر المحدد، ثم أظهرنا قيمة كل خلية منه على خانة نص.

الشفرة الخامسة: كيفية تغيير ألوان أسطر الداتاغريڤيو تناوبيا

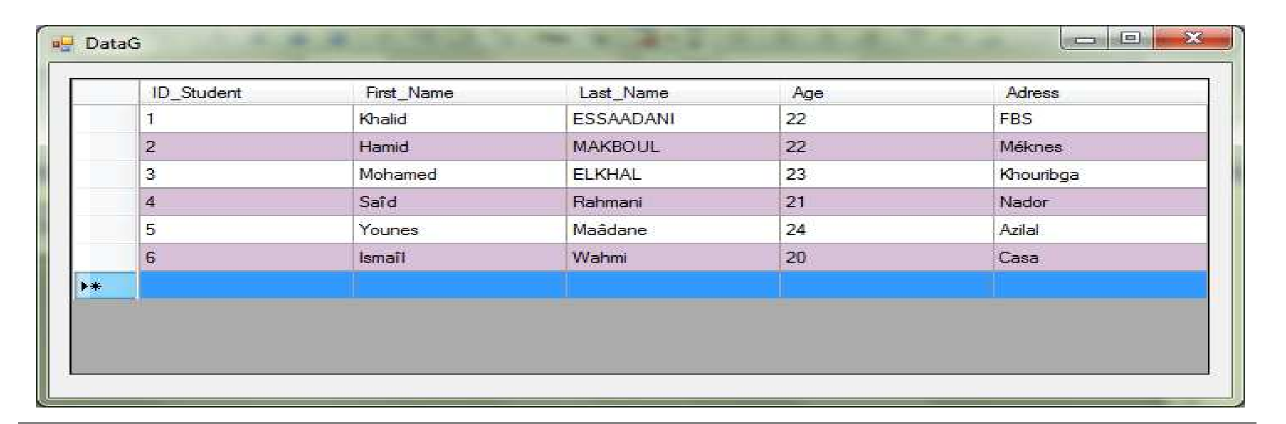
هذه المرة لن نكتب أي سطر من الكود، سنقوم فقط بالذهاب إلى خصائص الداتاغريديڤيو، ونحدد الخاصية AlternatingRowsDefaultCellStyle ، سوف تظهر لنا نافذة خصائص جديدة كما يلي:





سنختار منها الخاصية BackColor و نغيرها إلى اللون الذي نريد، وهذه صورة للنتيجة التي حصلت

عليها :



ID_Student	First_Name	Last_Name	Age	Adress
1	Khalid	ESSAADANI	22	FBS
2	Hamid	MAKBOUL	22	Méknes
3	Mohamed	ELKHAL	23	Khouribga
4	Saîd	Rahmani	21	Nador
5	Younes	Maâdane	24	Azilal
6	Ismaîl	Wahmi	20	Casa



الشفرة السادسة: كيفية البحث في خانة نص TextBox و إظهار النتائج في داتاغريدفيو

الحدث المرتبط بالتيكست بوكس، الذي يتولد عند عملية الكتابة يسمى TextChanged.

```
    Private Sub TextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox1.TextChanged

        Dt.Clear()
        Da = New OleDbDataAdapter("Select * From Student where
First_Name like '%" & TextBox1.Text & "%'", Con)
        Da.Fill(Dt)
        DataGridView1.DataSource = Dt

    End Sub
```

أعتقد أن كل شيء واضح ماعدا الجزء الأخير من الاستعلام:

```
Da = New OleDbDataAdapter("Select * From Student where
First_Name like '%" & TextBox1.Text & "%'", Con)
```

الكلمة like تعوض علامة تساوي "=" عندما نريد البحث عن بيانات لسنا متأكدين من تطابقها مع كلمة البحث، وتستعمل كلمة like مع رمز النسبة المئوية "%" الذي إذا وضعناه في الأول يعني: أن المهم من البحث هو ما وراءه، وإذا وضعناه في الأخير فيعني: أن المهم من البحث هو ما قبله، أما إذا قمنا بوضعه في الجهتين معا فيعني: أن المهم هو البحث عن كل كلمة تحتوي على الحروف المبحوث عنها مهما كانت البداية أو النهاية.



الشفرة السابعة: كيفية البحث العام (البحث عن أي كلمة في الجدول تتوافق نسبيا مع المبحوث عنه)

جواب هذا السؤال مثل الجواب السابق، الاختلاف سيكون فقط في الاستعلام:

```
    Private Sub TextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox1.TextChanged

        Dt.Clear()
        Da = New OleDbDataAdapter("Select * From Student where
First_Name+Last_Name+Adress like '%" & TextBox1.Text & "%'",
Con)
        Da.Fill(Dt)
        DataGridView1.DataSource = Dt

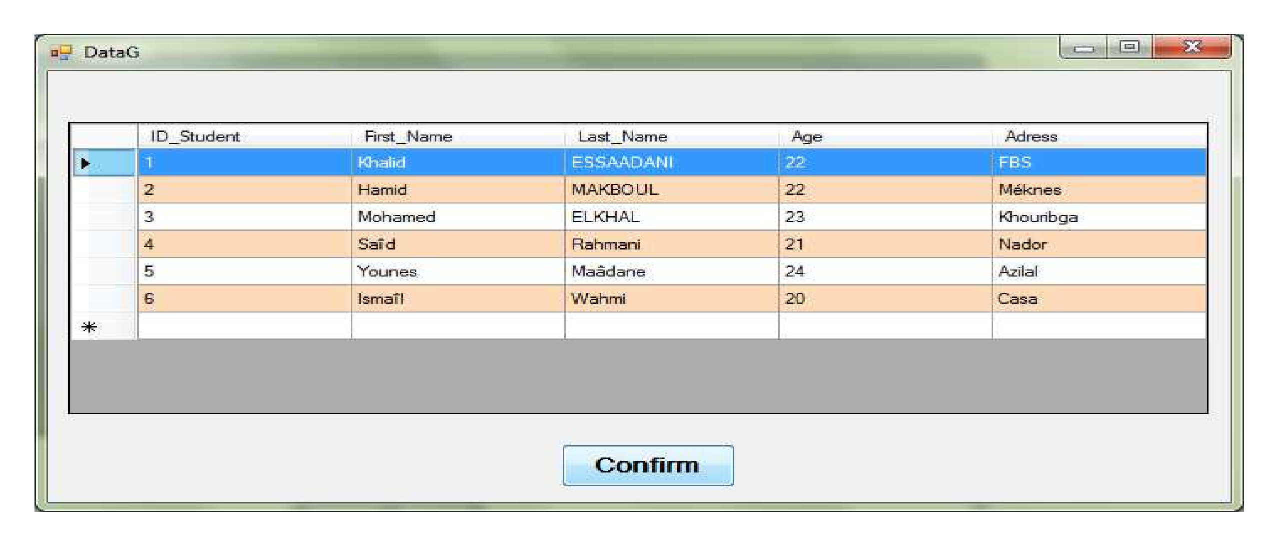
    End Sub
```

استعملنا عملية الدمج بين الحقول بالعلامة زائد "+" لجعل البحث متعدد وليس بواسطة حقل واحد.



الشفرة الثامنة: كيفية القيام بعمليات الإضافة والحذف والتعديل مباشرة على الداتاغريدفيو

وهذه من أجمل الشفرات التي تلخص كل العمليات التي رأيناها سابقا في سطرين فقط، سيكون التعديل والإضافة والحذف على الداتاغريدڤيو، وعند الانتهاء يقوم المستخدم بالضغط على زر التفعيل.



نضغط مرتين على الزر Confirm ونكتب الشفرة التالية:

```
    Private Sub Confirm_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Confirm.Click

        Cmdb = New OleDbCommandBuilder(Da)
        Da.Update(Dt)

    End Sub
```



تم والله الحمد



اللهم اجعل أعمالنا

خالصة لوجهك الكريم



الخاتمة:

نرجو من الله العلي القدير أن نكون قد وفقنا في أداء المهمة على أحسن وجه، ونتمنى أن يكون هذا العمل المتواضع قد أضاف إليك يا أخي الكريم ويا أختي الكريمة شيئا من المعلومات، ولا تنسوا أنها الخطى الأولى وما زال لنا معكم إن شاء الله خطوات، وفي كل خطوة سنحاول أن نرفع من مستوياتنا.

في الخطوة القادمة إن شاء الله سوف نتعرف على كيفية الربط بين الفيجوال استوديو و SQL Server بواسطة الإجراءات المخزنة Stored Procedures، مع كيفية إنشاء التقارير عن طريق Crystal Report

في انتظار الخطوة الثالثة من هذه السلسلة الضمنية "خطوة إلى الأمام" التابعة لسلسلة "كن أسدا"، تقبلوا منا أزكى الشكر، ولا تنسونا ووالدينا وكل المسلمين من صالح دعائكم، كما لا تنسونا من انتقاداتكم وتساؤلاتكم، وسنسعى قدر المستطاع إلى إجابتكم والأخذ بنصائحكم، وللمراسلة هذا عنواننا الإلكتروني: Khalid_ESSAADANI@Hotmail.fr

وهنا رقم هاتفي، فقط لمن كان على عجلة من أمره:

0673-07-51-05

أخوكم في الله: خالد السعداني